ANN-BS-HW04

Take 1000 images from CIFAR10. Then, given a pair of images from CIFAR10 $x1$ and $x2$, build a network that can return both images given their average as the only input.

Outline

Packages

In [1]:
```
import numpy as np
import pandas as pd
import tensorflow as tf
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, MaxPool2D, Flatten,BatchNormalizatio
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras import layers, models
from sklearn.metrics import confusion_matrix , classification_report
from keras.layers import AveragePooling2D
from keras.layers import GlobalMaxPooling2D
from keras.layers import GlobalAveragePooling2D
from tensorflow.keras.utils import to_categorical
import seaborn as sns
from keras.utils import plot_model
import random
from tensorflow.keras.models import Model
from tensorflow.keras import layers, losses
```

Loading Data

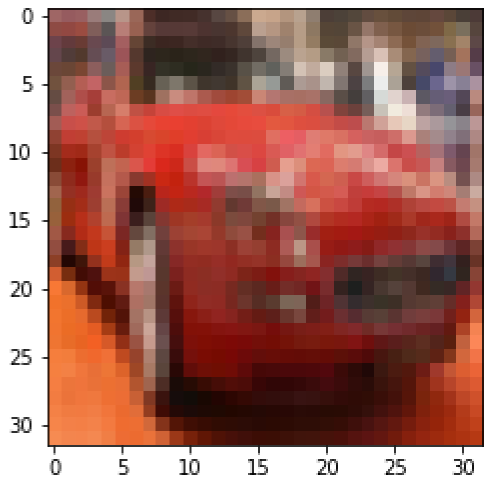
In [2]:
```
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.cifar10.load_data()
```

In [3]:
```
labels = ["airplane"
,"automobile"
, "bird"
, "cat"
,"deer"
,"dog"
,"frog"
,"horse"
,"ship"
,"truck"]
```

In [4]:

```
plt.imshow(x_train[5])
print(labels[int(y_train[5])])
```

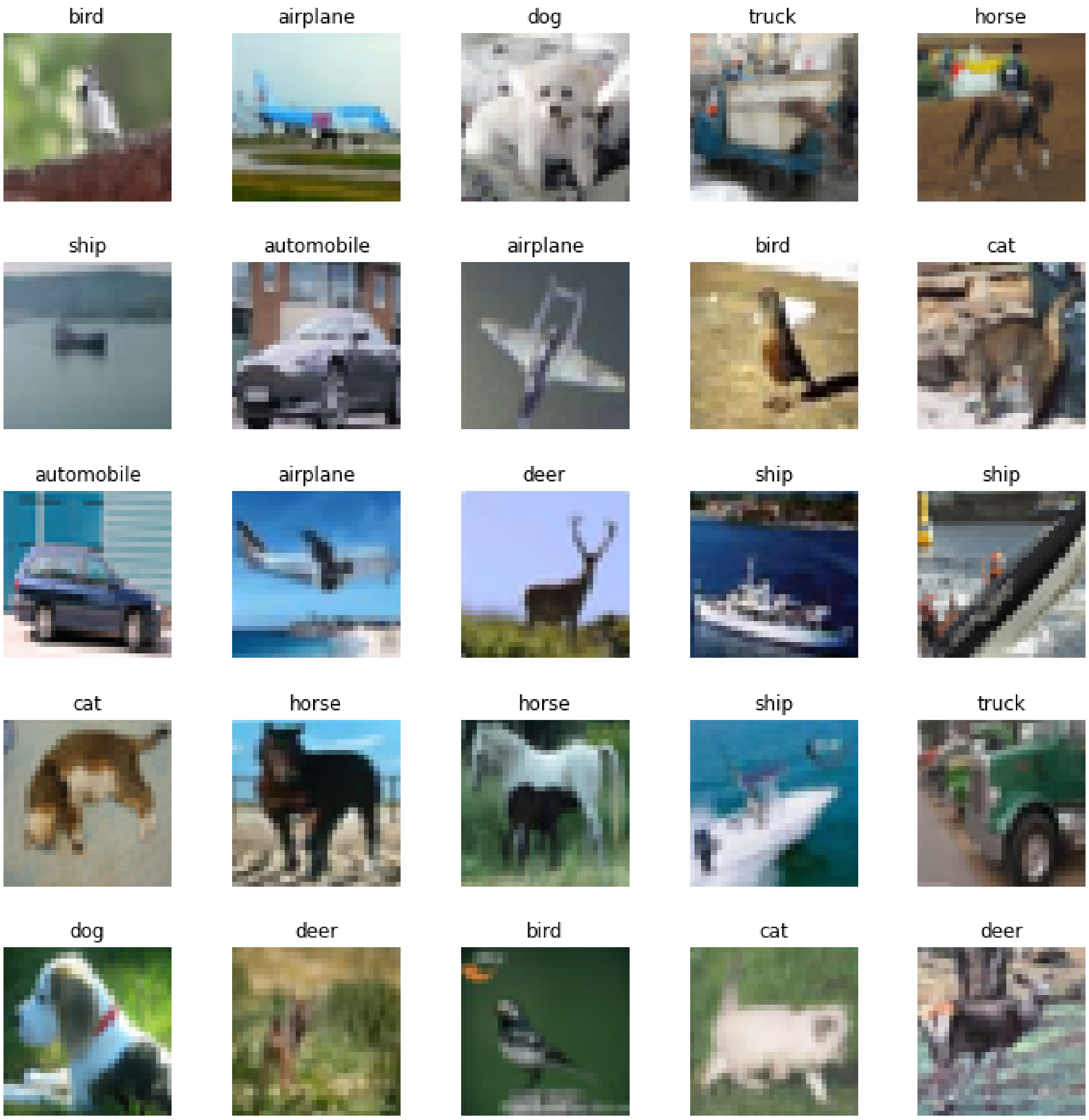
```
automobile
```



In [5]:

```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
X = x_train
fig, axes = plt.subplots(5,5, figsize=(10,10))
fig.tight_layout(pad=0.1)

for i,ax in enumerate(axes.flat):
    random_index = np.random.randint(50000)
    ax.imshow(X[random_index], cmap='gray')
    ax.set_title(labels[int(y_train[random_index])])
    ax.set_axis_off()
```



In [6]:
```
x_train.shape
```

Out[6]:
```
(50000, 32, 32, 3)
```

Building inputs

In [7]:
```
meanTrainingsetLabel = np.array([])
meanTrainingset = np.array([])
```

imageset1 for x1 / imageset2 for x2

In [8]:
```
imageset1 =  np.array([])
imageset2 =  np.array([])
```

Split data in two half and choose x1 from first one and x2 from second one

And normalize data

In [11]:
```
x_train = x_train/255
```

In [12]:
```
for i in range(1000):
    x1 = random.randint(0, 25000)
    x2 = random.randint(25000,50000)
    imageset1 = np.append(imageset1,x_train[x1])
    imageset2 = np.append(imageset2,x_train[x2])
    x3 = (x_train[x1] + x_train[x2])/2
    x3Label =labels[int(y_train[x1])]+"." +labels[int(y_train[x2])]
    meanTrainingsetLabel = np.append(meanTrainingsetLabel, x3Label)
    meanTrainingset = np.append(meanTrainingset, x3)
```

In [13]:
```
meanTrainingsetLabel[5]
```

Out[13]:
```
'airplane.ship'
```

In [14]:
```
meanTrainingset = meanTrainingset.reshape(1000,32,32,3)
```

Our input is ready

In [15]:
```
meanTrainingset.shape
```

Out[15]:
```
(1000, 32, 32, 3)
```

In [16]:
```
imageset1.shape
```

Out[16]:
```
(3072000,)
```

In [17]:
```
imageset1 = imageset1.reshape(1000,32,32,3)
```

In [18]:
```
imageset2 = imageset2.reshape(1000 ,32,32,3)
```

In [19]:
```
imageset2.shape
```

Out[19]:
```
(1000, 32, 32, 3)
```

In [20]:
```
meanTrainingset[2].shape
```

Out[20]:
```
(32, 32, 3)
```

Modeling

I'm going to use Normal Autoencoder with two decoders to fit with the x1 and x2 dataset

Autoencoder

In [21]:

```
from tensorflow.keras.models import Model
from tensorflow.keras import layers, losses
```

In [22]:

```
latent_dim = 64
class AutoencoderWithoutNorm1(Model):
    def __init__(self, latent_dim):
        super(AutoencoderWithoutNorm1, self).__init__()
        self.latent_dim = latent_dim
        #Our encoder(using cnn)
        self.encoder = tf.keras.Sequential([
              layers.Input(shape=(32, 32, 3)),
              layers.Conv2D(16, (3, 3), activation='relu', padding='same', strides=2),
              layers.Conv2D(8, (3, 3), activation='relu', padding='same', strides=2)

        ])
        self.decoderOne = tf.keras.Sequential([
            layers.Conv2DTranspose(8, kernel_size=3, strides=2, activation='relu', padd
            layers.Conv2DTranspose(16, kernel_size=3, strides=2, activation='relu', pad
            layers.Conv2D(3, kernel_size=(3, 3), activation='sigmoid', padding='same')
        ])
        self.decoderTwo = tf.keras.Sequential([
            layers.Conv2DTranspose(8, kernel_size=3, strides=2, activation='relu', padd
            layers.Conv2DTranspose(16, kernel_size=3, strides=2, activation='relu', pad
            layers.Conv2D(3, kernel_size=(3, 3), activation='sigmoid', padding='same')
        ])

    def call(self, x):
        encoded = self.encoder(x)
        decoded1 = self.decoderOne(encoded)
        decoded2 = self.decoderTwo(encoded)
        return decoded1,decoded2

autoencoderWithoutNorm1 = AutoencoderWithoutNorm1(latent_dim)
```

In [23]:

```
autoencoderWithoutNorm1.compile(optimizer='adam', metrics=["accuracy"], loss=losses.Mea
```

In [25]:

```
autoencoderWithoutNorm1.fit(meanTrainingset,[imageset1,imageset2],epochs=50)
```

```
Epoch 1/50
32/32 [==============================] - 1s 32ms/step - loss: 0.0667 - output_1_loss: 0.
0334 - output_2_loss: 0.0333 - output_1_accuracy: 0.5766 - output_2_accuracy: 0.5713
Epoch 2/50
32/32 [==============================] - 1s 36ms/step - loss: 0.0668 - output_1_loss: 0.
0334 - output_2_loss: 0.0334 - output_1_accuracy: 0.5771 - output_2_accuracy: 0.5734
Epoch 3/50
32/32 [==============================] - 1s 35ms/step - loss: 0.0669 - output_1_loss: 0.
```

```
0335 - output_2_loss: 0.0334 - output_1_accuracy: 0.5731 - output_2_accuracy: 0.5749
Epoch 4/50
32/32 [==============================] - 1s 36ms/step - loss: 0.0670 - output_1_loss: 0.
0336 - output_2_loss: 0.0334 - output_1_accuracy: 0.5724 - output_2_accuracy: 0.5741
Epoch 5/50
32/32 [==============================] - 1s 35ms/step - loss: 0.0668 - output_1_loss: 0.
0335 - output_2_loss: 0.0333 - output_1_accuracy: 0.5752 - output_2_accuracy: 0.5729
Epoch 6/50
32/32 [==============================] - 1s 35ms/step - loss: 0.0678 - output_1_loss: 0.
0338 - output_2_loss: 0.0340 - output_1_accuracy: 0.5741 - output_2_accuracy: 0.5697
Epoch 7/50
32/32 [==============================] - 1s 34ms/step - loss: 0.0673 - output_1_loss: 0.
0337 - output_2_loss: 0.0336 - output_1_accuracy: 0.5763 - output_2_accuracy: 0.5750
Epoch 8/50
32/32 [==============================] - 1s 35ms/step - loss: 0.0669 - output_1_loss: 0.
0335 - output_2_loss: 0.0334 - output_1_accuracy: 0.5733 - output_2_accuracy: 0.5776
Epoch 9/50
32/32 [==============================] - 1s 35ms/step - loss: 0.0672 - output_1_loss: 0.
0336 - output_2_loss: 0.0336 - output_1_accuracy: 0.5680 - output_2_accuracy: 0.5697
Epoch 10/50
32/32 [==============================] - 1s 36ms/step - loss: 0.0672 - output_1_loss: 0.
0336 - output_2_loss: 0.0335 - output_1_accuracy: 0.5748 - output_2_accuracy: 0.5777
Epoch 11/50
32/32 [==============================] - 1s 36ms/step - loss: 0.0669 - output_1_loss: 0.
0335 - output_2_loss: 0.0334 - output_1_accuracy: 0.5765 - output_2_accuracy: 0.5713
Epoch 12/50
32/32 [==============================] - 1s 37ms/step - loss: 0.0666 - output_1_loss: 0.
0333 - output_2_loss: 0.0333 - output_1_accuracy: 0.5721 - output_2_accuracy: 0.5740
Epoch 13/50
32/32 [==============================] - 1s 39ms/step - loss: 0.0670 - output_1_loss: 0.
0335 - output_2_loss: 0.0335 - output_1_accuracy: 0.5724 - output_2_accuracy: 0.5772
Epoch 14/50
32/32 [==============================] - 1s 36ms/step - loss: 0.0675 - output_1_loss: 0.
0338 - output_2_loss: 0.0337 - output_1_accuracy: 0.5631 - output_2_accuracy: 0.5748
Epoch 15/50
32/32 [==============================] - 1s 36ms/step - loss: 0.0669 - output_1_loss: 0.
0335 - output_2_loss: 0.0334 - output_1_accuracy: 0.5756 - output_2_accuracy: 0.5721
Epoch 16/50
32/32 [==============================] - 1s 36ms/step - loss: 0.0687 - output_1_loss: 0.
0342 - output_2_loss: 0.0345 - output_1_accuracy: 0.5716 - output_2_accuracy: 0.5734
Epoch 17/50
32/32 [==============================] - 1s 33ms/step - loss: 0.0673 - output_1_loss: 0.
0336 - output_2_loss: 0.0337 - output_1_accuracy: 0.5767 - output_2_accuracy: 0.5709
Epoch 18/50
32/32 [==============================] - 1s 34ms/step - loss: 0.0671 - output_1_loss: 0.
0336 - output_2_loss: 0.0336 - output_1_accuracy: 0.5761 - output_2_accuracy: 0.5742
Epoch 19/50
32/32 [==============================] - 1s 34ms/step - loss: 0.0670 - output_1_loss: 0.
0336 - output_2_loss: 0.0335 - output_1_accuracy: 0.5690 - output_2_accuracy: 0.5710
Epoch 20/50
32/32 [==============================] - 1s 34ms/step - loss: 0.0672 - output_1_loss: 0.
0336 - output_2_loss: 0.0336 - output_1_accuracy: 0.5769 - output_2_accuracy: 0.5711
Epoch 21/50
32/32 [==============================] - 1s 34ms/step - loss: 0.0668 - output_1_loss: 0.
0335 - output_2_loss: 0.0334 - output_1_accuracy: 0.5669 - output_2_accuracy: 0.5720
Epoch 22/50
32/32 [==============================] - 1s 34ms/step - loss: 0.0668 - output_1_loss: 0.
0334 - output_2_loss: 0.0334 - output_1_accuracy: 0.5784 - output_2_accuracy: 0.5766
Epoch 23/50
32/32 [==============================] - 1s 38ms/step - loss: 0.0672 - output_1_loss: 0.
```

```
0336 - output_2_loss: 0.0336 - output_1_accuracy: 0.5666 - output_2_accuracy: 0.5737
Epoch 24/50
32/32 [==============================] - 1s 42ms/step - loss: 0.0667 - output_1_loss: 0.
0334 - output_2_loss: 0.0333 - output_1_accuracy: 0.5755 - output_2_accuracy: 0.5752
Epoch 25/50
32/32 [==============================] - 1s 43ms/step - loss: 0.0667 - output_1_loss: 0.
0333 - output_2_loss: 0.0334 - output_1_accuracy: 0.5808 - output_2_accuracy: 0.5781
Epoch 26/50
32/32 [==============================] - 1s 43ms/step - loss: 0.0669 - output_1_loss: 0.
0335 - output_2_loss: 0.0334 - output_1_accuracy: 0.5652 - output_2_accuracy: 0.5755
Epoch 27/50
32/32 [==============================] - 1s 42ms/step - loss: 0.0668 - output_1_loss: 0.
0334 - output_2_loss: 0.0333 - output_1_accuracy: 0.5720 - output_2_accuracy: 0.5720
Epoch 28/50
32/32 [==============================] - 1s 43ms/step - loss: 0.0668 - output_1_loss: 0.
0335 - output_2_loss: 0.0333 - output_1_accuracy: 0.5717 - output_2_accuracy: 0.5704
Epoch 29/50
32/32 [==============================] - 1s 43ms/step - loss: 0.0664 - output_1_loss: 0.
0333 - output_2_loss: 0.0332 - output_1_accuracy: 0.5775 - output_2_accuracy: 0.5757
Epoch 30/50
32/32 [==============================] - 2s 48ms/step - loss: 0.0671 - output_1_loss: 0.
0336 - output_2_loss: 0.0335 - output_1_accuracy: 0.5737 - output_2_accuracy: 0.5728
Epoch 31/50
32/32 [==============================] - 1s 45ms/step - loss: 0.0671 - output_1_loss: 0.
0335 - output_2_loss: 0.0335 - output_1_accuracy: 0.5669 - output_2_accuracy: 0.5780
Epoch 32/50
32/32 [==============================] - 1s 45ms/step - loss: 0.0674 - output_1_loss: 0.
0337 - output_2_loss: 0.0338 - output_1_accuracy: 0.5680 - output_2_accuracy: 0.5766
Epoch 33/50
32/32 [==============================] - 2s 48ms/step - loss: 0.0667 - output_1_loss: 0.
0334 - output_2_loss: 0.0333 - output_1_accuracy: 0.5792 - output_2_accuracy: 0.5755
Epoch 34/50
32/32 [==============================] - 1s 42ms/step - loss: 0.0666 - output_1_loss: 0.
0333 - output_2_loss: 0.0333 - output_1_accuracy: 0.5717 - output_2_accuracy: 0.5772
Epoch 35/50
32/32 [==============================] - 1s 42ms/step - loss: 0.0668 - output_1_loss: 0.
0334 - output_2_loss: 0.0334 - output_1_accuracy: 0.5792 - output_2_accuracy: 0.5751
Epoch 36/50
32/32 [==============================] - 1s 42ms/step - loss: 0.0665 - output_1_loss: 0.
0333 - output_2_loss: 0.0332 - output_1_accuracy: 0.5690 - output_2_accuracy: 0.5721
Epoch 37/50
32/32 [==============================] - 1s 43ms/step - loss: 0.0677 - output_1_loss: 0.
0338 - output_2_loss: 0.0339 - output_1_accuracy: 0.5797 - output_2_accuracy: 0.5682
Epoch 38/50
32/32 [==============================] - 1s 42ms/step - loss: 0.0667 - output_1_loss: 0.
0334 - output_2_loss: 0.0333 - output_1_accuracy: 0.5735 - output_2_accuracy: 0.5793
Epoch 39/50
32/32 [==============================] - 1s 43ms/step - loss: 0.0670 - output_1_loss: 0.
0335 - output_2_loss: 0.0335 - output_1_accuracy: 0.5688 - output_2_accuracy: 0.5725
Epoch 40/50
32/32 [==============================] - 1s 43ms/step - loss: 0.0665 - output_1_loss: 0.
0333 - output_2_loss: 0.0332 - output_1_accuracy: 0.5807 - output_2_accuracy: 0.5715
Epoch 41/50
32/32 [==============================] - 1s 44ms/step - loss: 0.0668 - output_1_loss: 0.
0334 - output_2_loss: 0.0334 - output_1_accuracy: 0.5775 - output_2_accuracy: 0.5768
Epoch 42/50
32/32 [==============================] - 1s 43ms/step - loss: 0.0670 - output_1_loss: 0.
0335 - output_2_loss: 0.0335 - output_1_accuracy: 0.5734 - output_2_accuracy: 0.5766
Epoch 43/50
32/32 [==============================] - 1s 46ms/step - loss: 0.0668 - output_1_loss: 0.
```

```
0335 - output_2_loss: 0.0333 - output_1_accuracy: 0.5751 - output_2_accuracy: 0.5745
Epoch 44/50
32/32 [==============================] - 1s 44ms/step - loss: 0.0667 - output_1_loss: 0.
0334 - output_2_loss: 0.0334 - output_1_accuracy: 0.5760 - output_2_accuracy: 0.5716
Epoch 45/50
32/32 [==============================] - 1s 43ms/step - loss: 0.0671 - output_1_loss: 0.
0335 - output_2_loss: 0.0336 - output_1_accuracy: 0.5760 - output_2_accuracy: 0.5769
Epoch 46/50
32/32 [==============================] - 1s 43ms/step - loss: 0.0665 - output_1_loss: 0.
0332 - output_2_loss: 0.0333 - output_1_accuracy: 0.5768 - output_2_accuracy: 0.5714
Epoch 47/50
32/32 [==============================] - 1s 42ms/step - loss: 0.0665 - output_1_loss: 0.
0333 - output_2_loss: 0.0333 - output_1_accuracy: 0.5749 - output_2_accuracy: 0.5759
Epoch 48/50
32/32 [==============================] - 1s 44ms/step - loss: 0.0666 - output_1_loss: 0.
0333 - output_2_loss: 0.0332 - output_1_accuracy: 0.5700 - output_2_accuracy: 0.5733
Epoch 49/50
32/32 [==============================] - 1s 44ms/step - loss: 0.0663 - output_1_loss: 0.
0332 - output_2_loss: 0.0331 - output_1_accuracy: 0.5797 - output_2_accuracy: 0.5742
Epoch 50/50
32/32 [==============================] - 1s 45ms/step - loss: 0.0664 - output_1_loss: 0.
0333 - output_2_loss: 0.0332 - output_1_accuracy: 0.5739 - output_2_accuracy: 0.5746
```

Out[25]:

```
<keras.callbacks.History at 0x218cf3f6490>
```

loss: 0.0664 - output_1_loss: 0.0333 - output_2_loss: 0.0332 - output_1_accuracy: 0.5739 - output_2_accuracy: 0.5746

In [29]:

```python
latent_dim = 64
class Autoencoder2(Model):
    def __init__(self, latent_dim):
        super(Autoencoder2, self).__init__()
        self.latent_dim = latent_dim
        #Our encoder(using cnn)
        self.encoder = tf.keras.Sequential([
              layers.Flatten(),
              layers.Dense(latent_dim, activation='relu'),
              layers.Dense(32, activation='relu'),
              layers.Dense(16, activation='relu'),



        ])
        self.decoderOne = tf.keras.Sequential([

            layers.Dense(3072, activation='softmax'),
            layers.Reshape((32, 32,3)),

        ])
        self.decoderTwo = tf.keras.Sequential([
             layers.Dense(3072, activation='softmax'),
             layers.Reshape((32, 32,3)),
        ])

    def call(self, x):
        encoded = self.encoder(x)
        decoded1 = self.decoderOne(encoded)
        decoded2 = self.decoderTwo(encoded)
        return decoded1,decoded2
```

```
autoencoder2 = Autoencoder2(latent_dim)
```

In [30]:
```
autoencoder2.compile(optimizer='adam',metrics=["accuracy"], loss=losses.MeanSquaredErro
```

In [31]:
```
autoencoder2.fit(meanTrainingset,[imageset1,imageset2],epochs=50)
```

```
Epoch 1/50
32/32 [==============================] - 1s 4ms/step - loss: 0.5750 - output_1_loss: 0.2
905 - output_2_loss: 0.2845 - output_1_accuracy: 0.3326 - output_2_accuracy: 0.3374
Epoch 2/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5750 - output_1_loss: 0.2
905 - output_2_loss: 0.2845 - output_1_accuracy: 0.3331 - output_2_accuracy: 0.3405
Epoch 3/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5750 - output_1_loss: 0.2
905 - output_2_loss: 0.2845 - output_1_accuracy: 0.3362 - output_2_accuracy: 0.3417
Epoch 4/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5750 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.3473 - output_2_accuracy: 0.3457
Epoch 5/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5749 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.3772 - output_2_accuracy: 0.3712
Epoch 6/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5749 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4017 - output_2_accuracy: 0.3884
Epoch 7/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5749 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4167 - output_2_accuracy: 0.4034
Epoch 8/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5749 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4223 - output_2_accuracy: 0.4104
Epoch 9/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5749 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4273 - output_2_accuracy: 0.4168
Epoch 10/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5749 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4326 - output_2_accuracy: 0.4224
Epoch 11/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4398 - output_2_accuracy: 0.4308
Epoch 12/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4452 - output_2_accuracy: 0.4360
Epoch 13/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4423 - output_2_accuracy: 0.4338
Epoch 14/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4427 - output_2_accuracy: 0.4350
Epoch 15/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4430 - output_2_accuracy: 0.4354
Epoch 16/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4430 - output_2_accuracy: 0.4360
Epoch 17/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4422 - output_2_accuracy: 0.4355
```

```
Epoch 18/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4422 - output_2_accuracy: 0.4354
Epoch 19/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4416 - output_2_accuracy: 0.4355
Epoch 20/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4426 - output_2_accuracy: 0.4367
Epoch 21/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4410 - output_2_accuracy: 0.4350
Epoch 22/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4401 - output_2_accuracy: 0.4353
Epoch 23/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4404 - output_2_accuracy: 0.4352
Epoch 24/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4385 - output_2_accuracy: 0.4339
Epoch 25/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4393 - output_2_accuracy: 0.4346
Epoch 26/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4387 - output_2_accuracy: 0.4341
Epoch 27/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4374 - output_2_accuracy: 0.4335
Epoch 28/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4361 - output_2_accuracy: 0.4332
Epoch 29/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4365 - output_2_accuracy: 0.4344
Epoch 30/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4360 - output_2_accuracy: 0.4342
Epoch 31/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4349 - output_2_accuracy: 0.4336
Epoch 32/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4344 - output_2_accuracy: 0.4334
Epoch 33/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4337 - output_2_accuracy: 0.4325
Epoch 34/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4339 - output_2_accuracy: 0.4338
Epoch 35/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4329 - output_2_accuracy: 0.4321
Epoch 36/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4336 - output_2_accuracy: 0.4330
Epoch 37/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4320 - output_2_accuracy: 0.4321
```

```
Epoch 38/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4308 - output_2_accuracy: 0.4320
Epoch 39/50
32/32 [==============================] - 0s 5ms/step - loss: 0.5747 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4306 - output_2_accuracy: 0.4314
Epoch 40/50
32/32 [==============================] - 0s 5ms/step - loss: 0.5747 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4278 - output_2_accuracy: 0.4299
Epoch 41/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4296 - output_2_accuracy: 0.4308
Epoch 42/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5747 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4298 - output_2_accuracy: 0.4309
Epoch 43/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5747 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4296 - output_2_accuracy: 0.4310
Epoch 44/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5747 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4300 - output_2_accuracy: 0.4307
Epoch 45/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5747 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4282 - output_2_accuracy: 0.4300
Epoch 46/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5747 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4283 - output_2_accuracy: 0.4299
Epoch 47/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5747 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4277 - output_2_accuracy: 0.4297
Epoch 48/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5747 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4274 - output_2_accuracy: 0.4296
Epoch 49/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5747 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4264 - output_2_accuracy: 0.4289
Epoch 50/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5747 - output_1_loss: 0.2
904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4265 - output_2_accuracy: 0.4290
```

Out[31]:

```
<keras.callbacks.History at 0x218cf58bbe0>
```

Autoencoder 2 : loss: 0.5747 - output_1_loss: 0.2904 - output_2_loss: 0.2843 - output_1_accuracy: 0.4265 - output_2_accuracy: 0.4290

In [37]:

```python
latent_dim = 64
class Autoencoder2_1(Model):
    def __init__(self, latent_dim):
        super(Autoencoder2_1, self).__init__()
        self.latent_dim = latent_dim
        #Our encoder(using cnn)
        self.encoder = tf.keras.Sequential([
              layers.Flatten(),
              layers.Dense(latent_dim, activation='relu'),
              layers.Dense(32, activation='relu'),
              layers.Dense(16, activation='relu'),
        ])
        self.decoderOne = tf.keras.Sequential([
            layers.Dense(512,activation='relu'),
             layers.Dense(10, activation='relu'),
             layers.Dense(3072, activation='softmax'),

             layers.Reshape((32, 32,3)),

        ])
        self.decoderTwo = tf.keras.Sequential([
             layers.Dense(512,activation='relu'),
             layers.Dense(10, activation='relu'),
             layers.Dense(3072, activation='softmax'),
             layers.Reshape((32, 32,3)),
        ])

    def call(self, x):
        encoded = self.encoder(x)
        decoded1 = self.decoderOne(encoded)
        decoded2 = self.decoderTwo(encoded)
        return decoded1,decoded2

autoencoder2_1 = Autoencoder2_1(latent_dim)
```

In [38]:

```
autoencoder2_1.compile(optimizer='adam',metrics=["accuracy"], loss=losses.MeanSquaredEr
```

In [39]:

```
autoencoder2_1.fit(meanTrainingset,[imageset1,imageset2],epochs=50)
```

```
Epoch 1/50
32/32 [==============================] - 1s 4ms/step - loss: 0.5750 - output_1_loss: 0.2
905 - output_2_loss: 0.2845 - output_1_accuracy: 0.3423 - output_2_accuracy: 0.3385
Epoch 2/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5750 - output_1_loss: 0.2
905 - output_2_loss: 0.2845 - output_1_accuracy: 0.3709 - output_2_accuracy: 0.3572
Epoch 3/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5750 - output_1_loss: 0.2
905 - output_2_loss: 0.2845 - output_1_accuracy: 0.4053 - output_2_accuracy: 0.3805
Epoch 4/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5750 - output_1_loss: 0.2
905 - output_2_loss: 0.2845 - output_1_accuracy: 0.4311 - output_2_accuracy: 0.4011
Epoch 5/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5750 - output_1_loss: 0.2
905 - output_2_loss: 0.2845 - output_1_accuracy: 0.4463 - output_2_accuracy: 0.4139
Epoch 6/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5750 - output_1_loss: 0.2
905 - output_2_loss: 0.2845 - output_1_accuracy: 0.4540 - output_2_accuracy: 0.4179
Epoch 7/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5750 - output_1_loss: 0.2
905 - output_2_loss: 0.2845 - output_1_accuracy: 0.4576 - output_2_accuracy: 0.4184
Epoch 8/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5750 - output_1_loss: 0.2
905 - output_2_loss: 0.2845 - output_1_accuracy: 0.4532 - output_2_accuracy: 0.4111
Epoch 9/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5750 - output_1_loss: 0.2
905 - output_2_loss: 0.2845 - output_1_accuracy: 0.4453 - output_2_accuracy: 0.4020
Epoch 10/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5750 - output_1_loss: 0.2
905 - output_2_loss: 0.2845 - output_1_accuracy: 0.4326 - output_2_accuracy: 0.3855
```

```
Epoch 11/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5750 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4105 - output_2_accuracy: 0.3672
Epoch 12/50
32/32 [==============================] - 0s 5ms/step - loss: 0.5750 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.3750 - output_2_accuracy: 0.3528
Epoch 13/50
32/32 [==============================] - 0s 5ms/step - loss: 0.5750 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.3582 - output_2_accuracy: 0.3644
Epoch 14/50
32/32 [==============================] - 0s 5ms/step - loss: 0.5749 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.3969 - output_2_accuracy: 0.3855
Epoch 15/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5749 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4285 - output_2_accuracy: 0.4055
Epoch 16/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5749 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4431 - output_2_accuracy: 0.4167
Epoch 17/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5749 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4520 - output_2_accuracy: 0.4243
Epoch 18/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5749 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4567 - output_2_accuracy: 0.4278
Epoch 19/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5749 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4616 - output_2_accuracy: 0.4329
Epoch 20/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5749 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4650 - output_2_accuracy: 0.4352
Epoch 21/50
32/32 [==============================] - 0s 5ms/step - loss: 0.5749 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4696 - output_2_accuracy: 0.4395
Epoch 22/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5749 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4729 - output_2_accuracy: 0.4439
Epoch 23/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5749 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4770 - output_2_accuracy: 0.4483
Epoch 24/50
32/32 [==============================] - 0s 5ms/step - loss: 0.5749 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4797 - output_2_accuracy: 0.4525
Epoch 25/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5749 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4836 - output_2_accuracy: 0.4572
Epoch 26/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4860 - output_2_accuracy: 0.4606
Epoch 27/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
905 - output_2_loss: 0.2844 - output_1_accuracy: 0.4863 - output_2_accuracy: 0.4623
Epoch 28/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4831 - output_2_accuracy: 0.4626
Epoch 29/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4784 - output_2_accuracy: 0.4625
Epoch 30/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4796 - output_2_accuracy: 0.4628
```

```
Epoch 31/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4801 - output_2_accuracy: 0.4651
Epoch 32/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4822 - output_2_accuracy: 0.4652
Epoch 33/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4811 - output_2_accuracy: 0.4651
Epoch 34/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4808 - output_2_accuracy: 0.4662
Epoch 35/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4826 - output_2_accuracy: 0.4669
Epoch 36/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4803 - output_2_accuracy: 0.4680
Epoch 37/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4825 - output_2_accuracy: 0.4684
Epoch 38/50
32/32 [==============================] - 0s 5ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4828 - output_2_accuracy: 0.4689
Epoch 39/50
32/32 [==============================] - 0s 5ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4810 - output_2_accuracy: 0.4696
Epoch 40/50
32/32 [==============================] - 0s 5ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4824 - output_2_accuracy: 0.4715
Epoch 41/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4814 - output_2_accuracy: 0.4724
Epoch 42/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4823 - output_2_accuracy: 0.4736
Epoch 43/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4823 - output_2_accuracy: 0.4738
Epoch 44/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4815 - output_2_accuracy: 0.4748
Epoch 45/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4822 - output_2_accuracy: 0.4751
Epoch 46/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4849 - output_2_accuracy: 0.4768
Epoch 47/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4845 - output_2_accuracy: 0.4772
Epoch 48/50
32/32 [==============================] - 0s 5ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4841 - output_2_accuracy: 0.4775
Epoch 49/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4819 - output_2_accuracy: 0.4778
Epoch 50/50
32/32 [==============================] - 0s 4ms/step - loss: 0.5748 - output_1_loss: 0.2
904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4857 - output_2_accuracy: 0.4782
```

Out[39]:
```
<keras.callbacks.History at 0x218d15bbcd0>
```

Autoencoder 2_1 : loss: 0.5748 - output_1_loss: 0.2904 - output_2_loss: 0.2844 - output_1_accuracy: 0.4857 - output_2_accuracy: 0.4782.4290

In [ ]:
```

```

latent_dim = 32

In [40]:
```
latent_dim = 32
class Autoencoder32(Model):
    def __init__(self, latent_dim):
        super(Autoencoder32, self).__init__()
        self.latent_dim = latent_dim
        #Our encoder(using cnn)
        self.encoder = tf.keras.Sequential([
              layers.Input(shape=(32, 32, 3)),
              layers.Conv2D(16, (3, 3), activation='relu', padding='same', strides=2),
              layers.Conv2D(8, (3, 3), activation='relu', padding='same', strides=2)

        ])
        self.decoderOne = tf.keras.Sequential([
            layers.Conv2DTranspose(8, kernel_size=3, strides=2, activation='relu', padd
            layers.Conv2DTranspose(16, kernel_size=3, strides=2, activation='relu', pad
            layers.Conv2D(3, kernel_size=(3, 3), activation='sigmoid', padding='same')
        ])
        self.decoderTwo = tf.keras.Sequential([
            layers.Conv2DTranspose(8, kernel_size=3, strides=2, activation='relu', padd
            layers.Conv2DTranspose(16, kernel_size=3, strides=2, activation='relu', pad
            layers.Conv2D(3, kernel_size=(3, 3), activation='sigmoid', padding='same')
        ])

    def call(self, x):
        encoded = self.encoder(x)
        decoded1 = self.decoderOne(encoded)
        decoded2 = self.decoderTwo(encoded)
        return decoded1,decoded2

autoencoder32 = Autoencoder32(latent_dim)
```

In [41]:
```
autoencoder32.compile(optimizer='adam',metrics=["accuracy"], loss=losses.MeanSquaredErr
```

In [43]:
```
autoencoder32.fit(meanTrainingset,[imageset1,imageset2],epochs=50)
```

```
Epoch 1/50
32/32 [==============================] - 1s 34ms/step - loss: 0.0795 - output_1_loss: 0.
0401 - output_2_loss: 0.0394 - output_1_accuracy: 0.4690 - output_2_accuracy: 0.4653
Epoch 2/50
32/32 [==============================] - 1s 35ms/step - loss: 0.0781 - output_1_loss: 0.
0392 - output_2_loss: 0.0389 - output_1_accuracy: 0.4665 - output_2_accuracy: 0.4787
Epoch 3/50
32/32 [==============================] - 1s 34ms/step - loss: 0.0776 - output_1_loss: 0.
0388 - output_2_loss: 0.0388 - output_1_accuracy: 0.4679 - output_2_accuracy: 0.4692
Epoch 4/50
```

```
32/32 [==============================] - 1s 36ms/step - loss: 0.0760 - output_1_loss: 0.
0382 - output_2_loss: 0.0378 - output_1_accuracy: 0.4857 - output_2_accuracy: 0.4808
Epoch 5/50
32/32 [==============================] - 1s 36ms/step - loss: 0.0753 - output_1_loss: 0.
0377 - output_2_loss: 0.0376 - output_1_accuracy: 0.4883 - output_2_accuracy: 0.4858
Epoch 6/50
32/32 [==============================] - 1s 35ms/step - loss: 0.0750 - output_1_loss: 0.
0375 - output_2_loss: 0.0376 - output_1_accuracy: 0.5087 - output_2_accuracy: 0.4861
Epoch 7/50
32/32 [==============================] - 1s 35ms/step - loss: 0.0732 - output_1_loss: 0.
0366 - output_2_loss: 0.0366 - output_1_accuracy: 0.5299 - output_2_accuracy: 0.5086
Epoch 8/50
32/32 [==============================] - 1s 35ms/step - loss: 0.0727 - output_1_loss: 0.
0363 - output_2_loss: 0.0364 - output_1_accuracy: 0.5360 - output_2_accuracy: 0.5091
Epoch 9/50
32/32 [==============================] - 1s 36ms/step - loss: 0.0720 - output_1_loss: 0.
0359 - output_2_loss: 0.0361 - output_1_accuracy: 0.5477 - output_2_accuracy: 0.5299
Epoch 10/50
32/32 [==============================] - 1s 34ms/step - loss: 0.0713 - output_1_loss: 0.
0356 - output_2_loss: 0.0357 - output_1_accuracy: 0.5512 - output_2_accuracy: 0.5328
Epoch 11/50
32/32 [==============================] - 1s 38ms/step - loss: 0.0711 - output_1_loss: 0.
0356 - output_2_loss: 0.0355 - output_1_accuracy: 0.5558 - output_2_accuracy: 0.5432
Epoch 12/50
32/32 [==============================] - 1s 37ms/step - loss: 0.0713 - output_1_loss: 0.
0356 - output_2_loss: 0.0357 - output_1_accuracy: 0.5488 - output_2_accuracy: 0.5480
Epoch 13/50
32/32 [==============================] - 1s 36ms/step - loss: 0.0711 - output_1_loss: 0.
0355 - output_2_loss: 0.0356 - output_1_accuracy: 0.5542 - output_2_accuracy: 0.5442
Epoch 14/50
32/32 [==============================] - 1s 34ms/step - loss: 0.0702 - output_1_loss: 0.
0351 - output_2_loss: 0.0351 - output_1_accuracy: 0.5529 - output_2_accuracy: 0.5493
Epoch 15/50
32/32 [==============================] - 1s 35ms/step - loss: 0.0710 - output_1_loss: 0.
0354 - output_2_loss: 0.0355 - output_1_accuracy: 0.5543 - output_2_accuracy: 0.5491
Epoch 16/50
32/32 [==============================] - 1s 36ms/step - loss: 0.0700 - output_1_loss: 0.
0350 - output_2_loss: 0.0350 - output_1_accuracy: 0.5595 - output_2_accuracy: 0.5576
Epoch 17/50
32/32 [==============================] - 1s 35ms/step - loss: 0.0701 - output_1_loss: 0.
0351 - output_2_loss: 0.0350 - output_1_accuracy: 0.5563 - output_2_accuracy: 0.5591
Epoch 18/50
32/32 [==============================] - 1s 35ms/step - loss: 0.0697 - output_1_loss: 0.
0349 - output_2_loss: 0.0349 - output_1_accuracy: 0.5626 - output_2_accuracy: 0.5553
Epoch 19/50
32/32 [==============================] - 1s 35ms/step - loss: 0.0696 - output_1_loss: 0.
0348 - output_2_loss: 0.0348 - output_1_accuracy: 0.5550 - output_2_accuracy: 0.5538
Epoch 20/50
32/32 [==============================] - 1s 35ms/step - loss: 0.0692 - output_1_loss: 0.
0347 - output_2_loss: 0.0346 - output_1_accuracy: 0.5581 - output_2_accuracy: 0.5555
Epoch 21/50
32/32 [==============================] - 1s 34ms/step - loss: 0.0698 - output_1_loss: 0.
0349 - output_2_loss: 0.0349 - output_1_accuracy: 0.5595 - output_2_accuracy: 0.5616
Epoch 22/50
32/32 [==============================] - 1s 34ms/step - loss: 0.0693 - output_1_loss: 0.
0347 - output_2_loss: 0.0346 - output_1_accuracy: 0.5600 - output_2_accuracy: 0.5566
Epoch 23/50
32/32 [==============================] - 1s 38ms/step - loss: 0.0694 - output_1_loss: 0.
0347 - output_2_loss: 0.0347 - output_1_accuracy: 0.5597 - output_2_accuracy: 0.5611
Epoch 24/50
```

```
32/32 [==============================] - 1s 34ms/step - loss: 0.0689 - output_1_loss: 0.
0345 - output_2_loss: 0.0344 - output_1_accuracy: 0.5567 - output_2_accuracy: 0.5571
Epoch 25/50
32/32 [==============================] - 1s 36ms/step - loss: 0.0693 - output_1_loss: 0.
0346 - output_2_loss: 0.0346 - output_1_accuracy: 0.5638 - output_2_accuracy: 0.5627
Epoch 26/50
32/32 [==============================] - 1s 36ms/step - loss: 0.0692 - output_1_loss: 0.
0347 - output_2_loss: 0.0346 - output_1_accuracy: 0.5602 - output_2_accuracy: 0.5583
Epoch 27/50
32/32 [==============================] - 1s 35ms/step - loss: 0.0687 - output_1_loss: 0.
0344 - output_2_loss: 0.0343 - output_1_accuracy: 0.5600 - output_2_accuracy: 0.5645
Epoch 28/50
32/32 [==============================] - 1s 34ms/step - loss: 0.0693 - output_1_loss: 0.
0346 - output_2_loss: 0.0347 - output_1_accuracy: 0.5625 - output_2_accuracy: 0.5619
Epoch 29/50
32/32 [==============================] - 1s 37ms/step - loss: 0.0689 - output_1_loss: 0.
0344 - output_2_loss: 0.0344 - output_1_accuracy: 0.5638 - output_2_accuracy: 0.5652
Epoch 30/50
32/32 [==============================] - 1s 40ms/step - loss: 0.0694 - output_1_loss: 0.
0347 - output_2_loss: 0.0348 - output_1_accuracy: 0.5641 - output_2_accuracy: 0.5631
Epoch 31/50
32/32 [==============================] - 1s 40ms/step - loss: 0.0694 - output_1_loss: 0.
0347 - output_2_loss: 0.0347 - output_1_accuracy: 0.5615 - output_2_accuracy: 0.5610
Epoch 32/50
32/32 [==============================] - 1s 42ms/step - loss: 0.0686 - output_1_loss: 0.
0343 - output_2_loss: 0.0342 - output_1_accuracy: 0.5600 - output_2_accuracy: 0.5643
Epoch 33/50
32/32 [==============================] - 1s 43ms/step - loss: 0.0688 - output_1_loss: 0.
0344 - output_2_loss: 0.0344 - output_1_accuracy: 0.5635 - output_2_accuracy: 0.5666
Epoch 34/50
32/32 [==============================] - 1s 44ms/step - loss: 0.0684 - output_1_loss: 0.
0343 - output_2_loss: 0.0341 - output_1_accuracy: 0.5608 - output_2_accuracy: 0.5642
Epoch 35/50
32/32 [==============================] - 2s 50ms/step - loss: 0.0684 - output_1_loss: 0.
0343 - output_2_loss: 0.0342 - output_1_accuracy: 0.5655 - output_2_accuracy: 0.5592
Epoch 36/50
32/32 [==============================] - 2s 50ms/step - loss: 0.0685 - output_1_loss: 0.
0343 - output_2_loss: 0.0342 - output_1_accuracy: 0.5609 - output_2_accuracy: 0.5624
Epoch 37/50
32/32 [==============================] - 1s 40ms/step - loss: 0.0682 - output_1_loss: 0.
0342 - output_2_loss: 0.0341 - output_1_accuracy: 0.5632 - output_2_accuracy: 0.5607
Epoch 38/50
32/32 [==============================] - 1s 41ms/step - loss: 0.0683 - output_1_loss: 0.
0342 - output_2_loss: 0.0341 - output_1_accuracy: 0.5683 - output_2_accuracy: 0.5664
Epoch 39/50
32/32 [==============================] - 1s 41ms/step - loss: 0.0696 - output_1_loss: 0.
0347 - output_2_loss: 0.0349 - output_1_accuracy: 0.5624 - output_2_accuracy: 0.5623
Epoch 40/50
32/32 [==============================] - 1s 42ms/step - loss: 0.0683 - output_1_loss: 0.
0342 - output_2_loss: 0.0341 - output_1_accuracy: 0.5661 - output_2_accuracy: 0.5615
Epoch 41/50
32/32 [==============================] - 1s 40ms/step - loss: 0.0682 - output_1_loss: 0.
0341 - output_2_loss: 0.0341 - output_1_accuracy: 0.5629 - output_2_accuracy: 0.5670
Epoch 42/50
32/32 [==============================] - 1s 43ms/step - loss: 0.0684 - output_1_loss: 0.
0342 - output_2_loss: 0.0342 - output_1_accuracy: 0.5628 - output_2_accuracy: 0.5616
Epoch 43/50
32/32 [==============================] - 1s 41ms/step - loss: 0.0684 - output_1_loss: 0.
0343 - output_2_loss: 0.0342 - output_1_accuracy: 0.5667 - output_2_accuracy: 0.5591
Epoch 44/50
```

```
32/32 [==============================] - 1s 42ms/step - loss: 0.0682 - output_1_loss: 0.
0341 - output_2_loss: 0.0340 - output_1_accuracy: 0.5640 - output_2_accuracy: 0.5622
Epoch 45/50
32/32 [==============================] - 1s 41ms/step - loss: 0.0685 - output_1_loss: 0.
0343 - output_2_loss: 0.0342 - output_1_accuracy: 0.5675 - output_2_accuracy: 0.5593
Epoch 46/50
32/32 [==============================] - 1s 43ms/step - loss: 0.0681 - output_1_loss: 0.
0341 - output_2_loss: 0.0340 - output_1_accuracy: 0.5669 - output_2_accuracy: 0.5619
Epoch 47/50
32/32 [==============================] - 1s 42ms/step - loss: 0.0684 - output_1_loss: 0.
0342 - output_2_loss: 0.0342 - output_1_accuracy: 0.5642 - output_2_accuracy: 0.5683
Epoch 48/50
32/32 [==============================] - 1s 41ms/step - loss: 0.0681 - output_1_loss: 0.
0341 - output_2_loss: 0.0340 - output_1_accuracy: 0.5624 - output_2_accuracy: 0.5649
Epoch 49/50
32/32 [==============================] - 1s 41ms/step - loss: 0.0679 - output_1_loss: 0.
0340 - output_2_loss: 0.0339 - output_1_accuracy: 0.5688 - output_2_accuracy: 0.5602
Epoch 50/50
32/32 [==============================] - 1s 42ms/step - loss: 0.0690 - output_1_loss: 0.
0345 - output_2_loss: 0.0345 - output_1_accuracy: 0.5596 - output_2_accuracy: 0.5648
```

Out[43]:

```
<keras.callbacks.History at 0x218d4835130>
```

In [ ]:

In [ ]:

In [ ]:

latent_dim = 128

In [92]:

```python
latent_dim = 128
class Autoencoder128(Model):
    def __init__(self, latent_dim):
        super(Autoencoder128, self).__init__()
        self.latent_dim = latent_dim
        #Our encoder(using cnn)
        self.encoder = tf.keras.Sequential([
              layers.Flatten(),
              layers.Dense(latent_dim, activation='relu'),
        ])
        self.decoderOne = tf.keras.Sequential([
            layers.Dense(3072, activation='sigmoid'),
            layers.Reshape((32, 32,3))
        ])
        self.decoderTwo = tf.keras.Sequential([
            layers.Dense(3072, activation='sigmoid'),
            layers.Reshape((32, 32,3))
        ])

    def call(self, x):
        encoded = self.encoder(x)
        decoded1 = self.decoderOne(encoded)
        decoded2 = self.decoderTwo(encoded)
        return decoded1,decoded2
```

In [96]:
```
autoencoder128 = Autoencoder128(latent_dim)
```

In [97]:
```
autoencoder128.compile(optimizer='adam',metrics=["accuracy"], loss=losses.MeanSquaredEr
```

In [99]:
```
autoencoder128.fit(meanTrainingset,[imageset1,imageset2],epochs=100)
```

```
Epoch 1/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1116 - output_1_loss: 0.
0563 - output_2_loss: 0.0553 - output_1_accuracy: 0.5017 - output_2_accuracy: 0.4641
Epoch 2/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1114 - output_1_loss: 0.
0562 - output_2_loss: 0.0552 - output_1_accuracy: 0.4969 - output_2_accuracy: 0.4694
Epoch 3/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1109 - output_1_loss: 0.
0560 - output_2_loss: 0.0549 - output_1_accuracy: 0.4992 - output_2_accuracy: 0.4672
Epoch 4/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1109 - output_1_loss: 0.
0560 - output_2_loss: 0.0549 - output_1_accuracy: 0.5013 - output_2_accuracy: 0.4603
Epoch 5/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1107 - output_1_loss: 0.
0558 - output_2_loss: 0.0549 - output_1_accuracy: 0.5039 - output_2_accuracy: 0.4601
Epoch 6/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1104 - output_1_loss: 0.
0557 - output_2_loss: 0.0547 - output_1_accuracy: 0.5010 - output_2_accuracy: 0.4784
Epoch 7/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1103 - output_1_loss: 0.
0557 - output_2_loss: 0.0546 - output_1_accuracy: 0.5075 - output_2_accuracy: 0.4599
Epoch 8/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1101 - output_1_loss: 0.
0555 - output_2_loss: 0.0546 - output_1_accuracy: 0.5037 - output_2_accuracy: 0.4671
Epoch 9/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1100 - output_1_loss: 0.
0554 - output_2_loss: 0.0546 - output_1_accuracy: 0.4988 - output_2_accuracy: 0.4594
Epoch 10/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1102 - output_1_loss: 0.
0555 - output_2_loss: 0.0547 - output_1_accuracy: 0.4996 - output_2_accuracy: 0.4664
Epoch 11/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1098 - output_1_loss: 0.
0554 - output_2_loss: 0.0545 - output_1_accuracy: 0.5020 - output_2_accuracy: 0.4674
Epoch 12/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1096 - output_1_loss: 0.
0552 - output_2_loss: 0.0544 - output_1_accuracy: 0.5018 - output_2_accuracy: 0.4624
Epoch 13/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1095 - output_1_loss: 0.
0551 - output_2_loss: 0.0544 - output_1_accuracy: 0.4899 - output_2_accuracy: 0.4725
Epoch 14/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1093 - output_1_loss: 0.
0551 - output_2_loss: 0.0543 - output_1_accuracy: 0.5075 - output_2_accuracy: 0.4712
Epoch 15/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1093 - output_1_loss: 0.
0551 - output_2_loss: 0.0542 - output_1_accuracy: 0.4907 - output_2_accuracy: 0.4571
Epoch 16/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1094 - output_1_loss: 0.
0551 - output_2_loss: 0.0543 - output_1_accuracy: 0.4953 - output_2_accuracy: 0.4571
Epoch 17/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1090 - output_1_loss: 0.
```

```
0549 - output_2_loss: 0.0542 - output_1_accuracy: 0.5056 - output_2_accuracy: 0.4562
Epoch 18/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1091 - output_1_loss: 0.
0549 - output_2_loss: 0.0542 - output_1_accuracy: 0.4906 - output_2_accuracy: 0.4604
Epoch 19/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1089 - output_1_loss: 0.
0548 - output_2_loss: 0.0541 - output_1_accuracy: 0.4920 - output_2_accuracy: 0.4559
Epoch 20/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1087 - output_1_loss: 0.
0547 - output_2_loss: 0.0540 - output_1_accuracy: 0.4970 - output_2_accuracy: 0.4558
Epoch 21/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1085 - output_1_loss: 0.
0545 - output_2_loss: 0.0540 - output_1_accuracy: 0.4929 - output_2_accuracy: 0.4536
Epoch 22/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1086 - output_1_loss: 0.
0546 - output_2_loss: 0.0540 - output_1_accuracy: 0.4908 - output_2_accuracy: 0.4586
Epoch 23/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1085 - output_1_loss: 0.
0546 - output_2_loss: 0.0540 - output_1_accuracy: 0.4898 - output_2_accuracy: 0.4491
Epoch 24/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1084 - output_1_loss: 0.
0545 - output_2_loss: 0.0540 - output_1_accuracy: 0.4962 - output_2_accuracy: 0.4530
Epoch 25/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1081 - output_1_loss: 0.
0543 - output_2_loss: 0.0538 - output_1_accuracy: 0.4900 - output_2_accuracy: 0.4496
Epoch 26/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1082 - output_1_loss: 0.
0543 - output_2_loss: 0.0538 - output_1_accuracy: 0.4847 - output_2_accuracy: 0.4494
Epoch 27/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1082 - output_1_loss: 0.
0545 - output_2_loss: 0.0537 - output_1_accuracy: 0.4895 - output_2_accuracy: 0.4630
Epoch 28/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1079 - output_1_loss: 0.
0542 - output_2_loss: 0.0537 - output_1_accuracy: 0.4914 - output_2_accuracy: 0.4395
Epoch 29/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1078 - output_1_loss: 0.
0542 - output_2_loss: 0.0536 - output_1_accuracy: 0.4926 - output_2_accuracy: 0.4492
Epoch 30/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1077 - output_1_loss: 0.
0540 - output_2_loss: 0.0536 - output_1_accuracy: 0.4927 - output_2_accuracy: 0.4484
Epoch 31/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1077 - output_1_loss: 0.
0541 - output_2_loss: 0.0536 - output_1_accuracy: 0.4854 - output_2_accuracy: 0.4430
Epoch 32/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1080 - output_1_loss: 0.
0544 - output_2_loss: 0.0537 - output_1_accuracy: 0.4943 - output_2_accuracy: 0.4481
Epoch 33/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1074 - output_1_loss: 0.
0540 - output_2_loss: 0.0534 - output_1_accuracy: 0.4921 - output_2_accuracy: 0.4417
Epoch 34/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1074 - output_1_loss: 0.
0540 - output_2_loss: 0.0534 - output_1_accuracy: 0.4901 - output_2_accuracy: 0.4594
Epoch 35/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1072 - output_1_loss: 0.
0538 - output_2_loss: 0.0534 - output_1_accuracy: 0.4894 - output_2_accuracy: 0.4443
Epoch 36/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1074 - output_1_loss: 0.
0541 - output_2_loss: 0.0533 - output_1_accuracy: 0.4964 - output_2_accuracy: 0.4459
Epoch 37/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1073 - output_1_loss: 0.
```

```
0538 - output_2_loss: 0.0535 - output_1_accuracy: 0.4843 - output_2_accuracy: 0.4483
Epoch 38/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1074 - output_1_loss: 0.
0540 - output_2_loss: 0.0534 - output_1_accuracy: 0.4831 - output_2_accuracy: 0.4388
Epoch 39/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1073 - output_1_loss: 0.
0539 - output_2_loss: 0.0534 - output_1_accuracy: 0.4933 - output_2_accuracy: 0.4521
Epoch 40/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1071 - output_1_loss: 0.
0539 - output_2_loss: 0.0533 - output_1_accuracy: 0.4913 - output_2_accuracy: 0.4412
Epoch 41/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1068 - output_1_loss: 0.
0537 - output_2_loss: 0.0531 - output_1_accuracy: 0.4915 - output_2_accuracy: 0.4509
Epoch 42/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1073 - output_1_loss: 0.
0539 - output_2_loss: 0.0533 - output_1_accuracy: 0.4917 - output_2_accuracy: 0.4364
Epoch 43/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1068 - output_1_loss: 0.
0535 - output_2_loss: 0.0533 - output_1_accuracy: 0.4855 - output_2_accuracy: 0.4490
Epoch 44/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1067 - output_1_loss: 0.
0535 - output_2_loss: 0.0533 - output_1_accuracy: 0.4883 - output_2_accuracy: 0.4443
Epoch 45/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1067 - output_1_loss: 0.
0537 - output_2_loss: 0.0530 - output_1_accuracy: 0.4870 - output_2_accuracy: 0.4468
Epoch 46/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1064 - output_1_loss: 0.
0534 - output_2_loss: 0.0530 - output_1_accuracy: 0.4900 - output_2_accuracy: 0.4489
Epoch 47/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1064 - output_1_loss: 0.
0534 - output_2_loss: 0.0530 - output_1_accuracy: 0.4939 - output_2_accuracy: 0.4482
Epoch 48/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1064 - output_1_loss: 0.
0534 - output_2_loss: 0.0530 - output_1_accuracy: 0.4888 - output_2_accuracy: 0.4294
Epoch 49/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1063 - output_1_loss: 0.
0533 - output_2_loss: 0.0530 - output_1_accuracy: 0.4896 - output_2_accuracy: 0.4410
Epoch 50/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1063 - output_1_loss: 0.
0534 - output_2_loss: 0.0529 - output_1_accuracy: 0.4932 - output_2_accuracy: 0.4585
Epoch 51/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1060 - output_1_loss: 0.
0531 - output_2_loss: 0.0529 - output_1_accuracy: 0.5014 - output_2_accuracy: 0.4396
Epoch 52/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1064 - output_1_loss: 0.
0535 - output_2_loss: 0.0529 - output_1_accuracy: 0.4927 - output_2_accuracy: 0.4527
Epoch 53/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1064 - output_1_loss: 0.
0533 - output_2_loss: 0.0530 - output_1_accuracy: 0.4883 - output_2_accuracy: 0.4422
Epoch 54/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1060 - output_1_loss: 0.
0532 - output_2_loss: 0.0528 - output_1_accuracy: 0.4918 - output_2_accuracy: 0.4444
Epoch 55/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1059 - output_1_loss: 0.
0531 - output_2_loss: 0.0527 - output_1_accuracy: 0.4955 - output_2_accuracy: 0.4538
Epoch 56/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1061 - output_1_loss: 0.
0533 - output_2_loss: 0.0528 - output_1_accuracy: 0.4929 - output_2_accuracy: 0.4449
Epoch 57/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1058 - output_1_loss: 0.
```

```
0531 - output_2_loss: 0.0527 - output_1_accuracy: 0.4957 - output_2_accuracy: 0.4477
Epoch 58/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1056 - output_1_loss: 0.
0530 - output_2_loss: 0.0527 - output_1_accuracy: 0.4876 - output_2_accuracy: 0.4474
Epoch 59/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1060 - output_1_loss: 0.
0531 - output_2_loss: 0.0529 - output_1_accuracy: 0.4946 - output_2_accuracy: 0.4424
Epoch 60/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1062 - output_1_loss: 0.
0535 - output_2_loss: 0.0527 - output_1_accuracy: 0.4934 - output_2_accuracy: 0.4474
Epoch 61/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1056 - output_1_loss: 0.
0529 - output_2_loss: 0.0527 - output_1_accuracy: 0.4881 - output_2_accuracy: 0.4497
Epoch 62/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1057 - output_1_loss: 0.
0530 - output_2_loss: 0.0527 - output_1_accuracy: 0.4936 - output_2_accuracy: 0.4389
Epoch 63/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1055 - output_1_loss: 0.
0528 - output_2_loss: 0.0526 - output_1_accuracy: 0.4989 - output_2_accuracy: 0.4546
Epoch 64/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1053 - output_1_loss: 0.
0528 - output_2_loss: 0.0525 - output_1_accuracy: 0.4914 - output_2_accuracy: 0.4390
Epoch 65/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1053 - output_1_loss: 0.
0528 - output_2_loss: 0.0525 - output_1_accuracy: 0.4915 - output_2_accuracy: 0.4551
Epoch 66/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1052 - output_1_loss: 0.
0529 - output_2_loss: 0.0524 - output_1_accuracy: 0.4943 - output_2_accuracy: 0.4407
Epoch 67/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1051 - output_1_loss: 0.
0528 - output_2_loss: 0.0523 - output_1_accuracy: 0.4987 - output_2_accuracy: 0.4453
Epoch 68/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1051 - output_1_loss: 0.
0528 - output_2_loss: 0.0523 - output_1_accuracy: 0.4935 - output_2_accuracy: 0.4566
Epoch 69/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1050 - output_1_loss: 0.
0527 - output_2_loss: 0.0523 - output_1_accuracy: 0.4835 - output_2_accuracy: 0.4255
Epoch 70/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1054 - output_1_loss: 0.
0529 - output_2_loss: 0.0525 - output_1_accuracy: 0.4998 - output_2_accuracy: 0.4552
Epoch 71/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1050 - output_1_loss: 0.
0527 - output_2_loss: 0.0523 - output_1_accuracy: 0.4968 - output_2_accuracy: 0.4517
Epoch 72/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1050 - output_1_loss: 0.
0527 - output_2_loss: 0.0523 - output_1_accuracy: 0.4900 - output_2_accuracy: 0.4480
Epoch 73/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1050 - output_1_loss: 0.
0527 - output_2_loss: 0.0523 - output_1_accuracy: 0.4882 - output_2_accuracy: 0.4535
Epoch 74/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1053 - output_1_loss: 0.
0531 - output_2_loss: 0.0522 - output_1_accuracy: 0.4924 - output_2_accuracy: 0.4540
Epoch 75/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1052 - output_1_loss: 0.
0529 - output_2_loss: 0.0523 - output_1_accuracy: 0.4959 - output_2_accuracy: 0.4489
Epoch 76/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1046 - output_1_loss: 0.
0525 - output_2_loss: 0.0521 - output_1_accuracy: 0.4886 - output_2_accuracy: 0.4552
Epoch 77/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1046 - output_1_loss: 0.
```

```
0525 - output_2_loss: 0.0520 - output_1_accuracy: 0.5012 - output_2_accuracy: 0.4490
Epoch 78/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1045 - output_1_loss: 0.
0523 - output_2_loss: 0.0522 - output_1_accuracy: 0.4940 - output_2_accuracy: 0.4488
Epoch 79/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1046 - output_1_loss: 0.
0526 - output_2_loss: 0.0520 - output_1_accuracy: 0.4924 - output_2_accuracy: 0.4457
Epoch 80/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1046 - output_1_loss: 0.
0524 - output_2_loss: 0.0523 - output_1_accuracy: 0.4924 - output_2_accuracy: 0.4495
Epoch 81/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1048 - output_1_loss: 0.
0524 - output_2_loss: 0.0524 - output_1_accuracy: 0.4959 - output_2_accuracy: 0.4512
Epoch 82/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1053 - output_1_loss: 0.
0527 - output_2_loss: 0.0527 - output_1_accuracy: 0.4895 - output_2_accuracy: 0.4508
Epoch 83/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1044 - output_1_loss: 0.
0524 - output_2_loss: 0.0520 - output_1_accuracy: 0.4875 - output_2_accuracy: 0.4523
Epoch 84/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1041 - output_1_loss: 0.
0522 - output_2_loss: 0.0519 - output_1_accuracy: 0.4954 - output_2_accuracy: 0.4415
Epoch 85/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1043 - output_1_loss: 0.
0523 - output_2_loss: 0.0520 - output_1_accuracy: 0.4950 - output_2_accuracy: 0.4495
Epoch 86/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1040 - output_1_loss: 0.
0522 - output_2_loss: 0.0518 - output_1_accuracy: 0.4964 - output_2_accuracy: 0.4567
Epoch 87/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1043 - output_1_loss: 0.
0524 - output_2_loss: 0.0518 - output_1_accuracy: 0.4829 - output_2_accuracy: 0.4480
Epoch 88/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1042 - output_1_loss: 0.
0522 - output_2_loss: 0.0520 - output_1_accuracy: 0.4956 - output_2_accuracy: 0.4570
Epoch 89/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1039 - output_1_loss: 0.
0520 - output_2_loss: 0.0518 - output_1_accuracy: 0.4967 - output_2_accuracy: 0.4474
Epoch 90/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1038 - output_1_loss: 0.
0521 - output_2_loss: 0.0517 - output_1_accuracy: 0.4850 - output_2_accuracy: 0.4584
Epoch 91/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1038 - output_1_loss: 0.
0521 - output_2_loss: 0.0517 - output_1_accuracy: 0.4907 - output_2_accuracy: 0.4503
Epoch 92/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1037 - output_1_loss: 0.
0521 - output_2_loss: 0.0516 - output_1_accuracy: 0.4964 - output_2_accuracy: 0.4609
Epoch 93/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1043 - output_1_loss: 0.
0523 - output_2_loss: 0.0520 - output_1_accuracy: 0.4942 - output_2_accuracy: 0.4433
Epoch 94/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1039 - output_1_loss: 0.
0520 - output_2_loss: 0.0519 - output_1_accuracy: 0.4930 - output_2_accuracy: 0.4561
Epoch 95/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1045 - output_1_loss: 0.
0523 - output_2_loss: 0.0522 - output_1_accuracy: 0.4897 - output_2_accuracy: 0.4582
Epoch 96/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1035 - output_1_loss: 0.
0519 - output_2_loss: 0.0516 - output_1_accuracy: 0.4961 - output_2_accuracy: 0.4526
Epoch 97/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1038 - output_1_loss: 0.
```

```
0520 - output_2_loss: 0.0517 - output_1_accuracy: 0.4877 - output_2_accuracy: 0.4459
Epoch 98/100
32/32 [==============================] - 0s 12ms/step - loss: 0.1042 - output_1_loss: 0.
0525 - output_2_loss: 0.0517 - output_1_accuracy: 0.4953 - output_2_accuracy: 0.4597
Epoch 99/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1034 - output_1_loss: 0.
0520 - output_2_loss: 0.0514 - output_1_accuracy: 0.4962 - output_2_accuracy: 0.4556
Epoch 100/100
32/32 [==============================] - 0s 11ms/step - loss: 0.1036 - output_1_loss: 0.
0519 - output_2_loss: 0.0517 - output_1_accuracy: 0.4860 - output_2_accuracy: 0.4544
```

Out[99]:
```
<keras.callbacks.History at 0x1770c8753d0>
```

Autoencoder32 : loss: 0.1036 - output_1_loss: 0.0519 - output_2_loss: 0.0517 - output_1_accuracy: 0.4860 - output_2_accuracy: 0.4544

In [56]:
```python
latent_dim = 64
class Autoencoder4(Model):
    def __init__(self, latent_dim):
        super(Autoencoder4, self).__init__()
        self.latent_dim = latent_dim
        #Our encoder(using cnn)
        self.encoder = tf.keras.Sequential([
              layers.Conv2D(32, kernel_size=3, strides=1, padding='same', activation='r
              BatchNormalization(),
              layers.Conv2D(32, kernel_size=3, strides=2, padding='same', activation='r
              layers.Conv2D(32, kernel_size=3, strides=1, padding='same', activation='r
              BatchNormalization(),


        ])
        self.decoderOne = tf.keras.Sequential([
            layers.UpSampling2D((2, 2)),
            layers.Conv2D(32, kernel_size=3, strides=1, padding='same', activation='rel
             BatchNormalization(),
            layers.Conv2D(3,  kernel_size=1, strides=1, padding='same', activation='sig

        ])
        self.decoderTwo = tf.keras.Sequential([
           layers.UpSampling2D((2, 2)),
            layers.Conv2D(32, kernel_size=3, strides=1, padding='same', activation='rel
             BatchNormalization(),
            layers.Conv2D(3,  kernel_size=1, strides=1, padding='same', activation='sig
        ])

    def call(self, x):
        encoded = self.encoder(x)
        decoded1 = self.decoderOne(encoded)
        decoded2 = self.decoderTwo(encoded)
        return decoded1,decoded2

autoencoder4= Autoencoder4(latent_dim)
```

In [57]:
```python
autoencoder4.compile(optimizer='adam',metrics=["accuracy"], loss=losses.MeanSquaredErro
In [58]: autoencoder4.fit(meanTrainingset,[imageset1,imageset2],epochs=100)
```

```
Epoch 1/100
32/32 [==============================] - 4s 88ms/step - loss: 0.1043 - output_1_loss: 0.
0526 - output_2_loss: 0.0517 - output_1_accuracy: 0.4175 - output_2_accuracy: 0.4148
Epoch 2/100
32/32 [==============================] - 3s 90ms/step - loss: 0.0733 - output_1_loss: 0.
0367 - output_2_loss: 0.0366 - output_1_accuracy: 0.4889 - output_2_accuracy: 0.5069
Epoch 3/100
32/32 [==============================] - 3s 91ms/step - loss: 0.0695 - output_1_loss: 0.
0348 - output_2_loss: 0.0347 - output_1_accuracy: 0.5333 - output_2_accuracy: 0.5338
Epoch 4/100
32/32 [==============================] - 4s 111ms/step - loss: 0.0680 - output_1_loss:
0.0339 - output_2_loss: 0.0341 - output_1_accuracy: 0.5602 - output_2_accuracy: 0.5539
Epoch 5/100
32/32 [==============================] - 4s 121ms/step - loss: 0.0673 - output_1_loss:
0.0336 - output_2_loss: 0.0337 - output_1_accuracy: 0.5701 - output_2_accuracy: 0.5642
Epoch 6/100
32/32 [==============================] - 4s 114ms/step - loss: 0.0665 - output_1_loss:
0.0332 - output_2_loss: 0.0333 - output_1_accuracy: 0.5723 - output_2_accuracy: 0.5786
Epoch 7/100
32/32 [==============================] - 3s 108ms/step - loss: 0.0661 - output_1_loss:
0.0330 - output_2_loss: 0.0331 - output_1_accuracy: 0.5874 - output_2_accuracy: 0.5827
Epoch 8/100
32/32 [==============================] - 3s 108ms/step - loss: 0.0659 - output_1_loss:
0.0329 - output_2_loss: 0.0330 - output_1_accuracy: 0.5885 - output_2_accuracy: 0.5902
Epoch 9/100
32/32 [==============================] - 3s 107ms/step - loss: 0.0656 - output_1_loss:
0.0327 - output_2_loss: 0.0329 - output_1_accuracy: 0.5951 - output_2_accuracy: 0.5947
Epoch 10/100
32/32 [==============================] - 3s 109ms/step - loss: 0.0654 - output_1_loss:
0.0329 - output_2_loss: 0.0326 - output_1_accuracy: 0.5991 - output_2_accuracy: 0.6066
Epoch 11/100
32/32 [==============================] - 3s 106ms/step - loss: 0.0652 - output_1_loss:
0.0325 - output_2_loss: 0.0327 - output_1_accuracy: 0.6062 - output_2_accuracy: 0.5950
Epoch 12/100
32/32 [==============================] - 3s 109ms/step - loss: 0.0649 - output_1_loss:
0.0323 - output_2_loss: 0.0326 - output_1_accuracy: 0.6049 - output_2_accuracy: 0.6116
Epoch 13/100
32/32 [==============================] - 4s 111ms/step - loss: 0.0649 - output_1_loss:
0.0324 - output_2_loss: 0.0325 - output_1_accuracy: 0.6046 - output_2_accuracy: 0.6123
Epoch 14/100
32/32 [==============================] - 4s 110ms/step - loss: 0.0648 - output_1_loss:
0.0325 - output_2_loss: 0.0323 - output_1_accuracy: 0.6081 - output_2_accuracy: 0.6105
Epoch 15/100
32/32 [==============================] - 4s 113ms/step - loss: 0.0646 - output_1_loss:
0.0324 - output_2_loss: 0.0322 - output_1_accuracy: 0.6116 - output_2_accuracy: 0.6095
Epoch 16/100
32/32 [==============================] - 3s 108ms/step - loss: 0.0650 - output_1_loss:
0.0324 - output_2_loss: 0.0326 - output_1_accuracy: 0.6095 - output_2_accuracy: 0.6183
Epoch 17/100
32/32 [==============================] - 3s 109ms/step - loss: 0.0643 - output_1_loss:
0.0321 - output_2_loss: 0.0322 - output_1_accuracy: 0.6116 - output_2_accuracy: 0.6133
Epoch 18/100
32/32 [==============================] - 4s 111ms/step - loss: 0.0642 - output_1_loss:
0.0320 - output_2_loss: 0.0322 - output_1_accuracy: 0.6202 - output_2_accuracy: 0.6189
Epoch 19/100
32/32 [==============================] - 4s 119ms/step - loss: 0.0646 - output_1_loss:
```

```
0.0323 - output_2_loss: 0.0323 - output_1_accuracy: 0.6133 - output_2_accuracy: 0.6202
Epoch 20/100
32/32 [==============================] - 4s 129ms/step - loss: 0.0641 - output_1_loss:
0.0320 - output_2_loss: 0.0322 - output_1_accuracy: 0.6176 - output_2_accuracy: 0.6208
Epoch 21/100
32/32 [==============================] - 4s 117ms/step - loss: 0.0641 - output_1_loss:
0.0320 - output_2_loss: 0.0321 - output_1_accuracy: 0.6228 - output_2_accuracy: 0.6161
Epoch 22/100
32/32 [==============================] - 4s 117ms/step - loss: 0.0643 - output_1_loss:
0.0320 - output_2_loss: 0.0323 - output_1_accuracy: 0.6227 - output_2_accuracy: 0.6223
Epoch 23/100
32/32 [==============================] - 3s 106ms/step - loss: 0.0640 - output_1_loss:
0.0319 - output_2_loss: 0.0321 - output_1_accuracy: 0.6246 - output_2_accuracy: 0.6251
Epoch 24/100
32/32 [==============================] - 4s 114ms/step - loss: 0.0643 - output_1_loss:
0.0321 - output_2_loss: 0.0322 - output_1_accuracy: 0.6215 - output_2_accuracy: 0.6158
Epoch 25/100
32/32 [==============================] - 3s 109ms/step - loss: 0.0639 - output_1_loss:
0.0320 - output_2_loss: 0.0319 - output_1_accuracy: 0.6212 - output_2_accuracy: 0.6231
Epoch 26/100
32/32 [==============================] - 4s 122ms/step - loss: 0.0638 - output_1_loss:
0.0320 - output_2_loss: 0.0317 - output_1_accuracy: 0.6297 - output_2_accuracy: 0.6259
Epoch 27/100
32/32 [==============================] - 4s 113ms/step - loss: 0.0642 - output_1_loss:
0.0322 - output_2_loss: 0.0320 - output_1_accuracy: 0.6227 - output_2_accuracy: 0.6275
Epoch 28/100
32/32 [==============================] - 4s 112ms/step - loss: 0.0641 - output_1_loss:
0.0322 - output_2_loss: 0.0319 - output_1_accuracy: 0.6228 - output_2_accuracy: 0.6258
Epoch 29/100
32/32 [==============================] - 4s 115ms/step - loss: 0.0636 - output_1_loss:
0.0318 - output_2_loss: 0.0317 - output_1_accuracy: 0.6220 - output_2_accuracy: 0.6259
Epoch 30/100
32/32 [==============================] - 4s 111ms/step - loss: 0.0638 - output_1_loss:
0.0318 - output_2_loss: 0.0320 - output_1_accuracy: 0.6233 - output_2_accuracy: 0.6287
Epoch 31/100
32/32 [==============================] - 4s 114ms/step - loss: 0.0637 - output_1_loss:
0.0319 - output_2_loss: 0.0317 - output_1_accuracy: 0.6274 - output_2_accuracy: 0.6238
Epoch 32/100
32/32 [==============================] - 4s 110ms/step - loss: 0.0636 - output_1_loss:
0.0318 - output_2_loss: 0.0318 - output_1_accuracy: 0.6252 - output_2_accuracy: 0.6287
Epoch 33/100
32/32 [==============================] - 4s 109ms/step - loss: 0.0635 - output_1_loss:
0.0316 - output_2_loss: 0.0319 - output_1_accuracy: 0.6286 - output_2_accuracy: 0.6303
Epoch 34/100
32/32 [==============================] - 3s 109ms/step - loss: 0.0634 - output_1_loss:
0.0318 - output_2_loss: 0.0315 - output_1_accuracy: 0.6276 - output_2_accuracy: 0.6311
Epoch 35/100
32/32 [==============================] - 3s 108ms/step - loss: 0.0634 - output_1_loss:
0.0318 - output_2_loss: 0.0317 - output_1_accuracy: 0.6295 - output_2_accuracy: 0.6315
Epoch 36/100
32/32 [==============================] - 3s 109ms/step - loss: 0.0634 - output_1_loss:
0.0317 - output_2_loss: 0.0317 - output_1_accuracy: 0.6342 - output_2_accuracy: 0.6320
Epoch 37/100
32/32 [==============================] - 3s 109ms/step - loss: 0.0637 - output_1_loss:
0.0317 - output_2_loss: 0.0319 - output_1_accuracy: 0.6325 - output_2_accuracy: 0.6313
Epoch 38/100
32/32 [==============================] - 3s 107ms/step - loss: 0.0637 - output_1_loss:
0.0318 - output_2_loss: 0.0319 - output_1_accuracy: 0.6262 - output_2_accuracy: 0.6294
Epoch 39/100
32/32 [==============================] - 3s 107ms/step - loss: 0.0636 - output_1_loss:
```

```
0.0318 - output_2_loss: 0.0318 - output_1_accuracy: 0.6315 - output_2_accuracy: 0.6320
Epoch 40/100
32/32 [==============================] - 3s 109ms/step - loss: 0.0633 - output_1_loss:
0.0318 - output_2_loss: 0.0315 - output_1_accuracy: 0.6329 - output_2_accuracy: 0.6402
Epoch 41/100
32/32 [==============================] - 3s 108ms/step - loss: 0.0632 - output_1_loss:
0.0317 - output_2_loss: 0.0315 - output_1_accuracy: 0.6313 - output_2_accuracy: 0.6342
Epoch 42/100
32/32 [==============================] - 3s 109ms/step - loss: 0.0633 - output_1_loss:
0.0316 - output_2_loss: 0.0316 - output_1_accuracy: 0.6321 - output_2_accuracy: 0.6329
Epoch 43/100
32/32 [==============================] - 3s 108ms/step - loss: 0.0636 - output_1_loss:
0.0316 - output_2_loss: 0.0320 - output_1_accuracy: 0.6323 - output_2_accuracy: 0.6319
Epoch 44/100
32/32 [==============================] - 3s 108ms/step - loss: 0.0637 - output_1_loss:
0.0317 - output_2_loss: 0.0321 - output_1_accuracy: 0.6296 - output_2_accuracy: 0.6283
Epoch 45/100
32/32 [==============================] - 3s 107ms/step - loss: 0.0632 - output_1_loss:
0.0316 - output_2_loss: 0.0316 - output_1_accuracy: 0.6344 - output_2_accuracy: 0.6387
Epoch 46/100
32/32 [==============================] - 3s 106ms/step - loss: 0.0633 - output_1_loss:
0.0318 - output_2_loss: 0.0315 - output_1_accuracy: 0.6334 - output_2_accuracy: 0.6335
Epoch 47/100
32/32 [==============================] - 3s 107ms/step - loss: 0.0633 - output_1_loss:
0.0318 - output_2_loss: 0.0315 - output_1_accuracy: 0.6267 - output_2_accuracy: 0.6344
Epoch 48/100
32/32 [==============================] - 4s 111ms/step - loss: 0.0634 - output_1_loss:
0.0318 - output_2_loss: 0.0316 - output_1_accuracy: 0.6322 - output_2_accuracy: 0.6303
Epoch 49/100
32/32 [==============================] - 3s 108ms/step - loss: 0.0632 - output_1_loss:
0.0318 - output_2_loss: 0.0315 - output_1_accuracy: 0.6412 - output_2_accuracy: 0.6329
Epoch 50/100
32/32 [==============================] - 3s 107ms/step - loss: 0.0630 - output_1_loss:
0.0314 - output_2_loss: 0.0316 - output_1_accuracy: 0.6275 - output_2_accuracy: 0.6367
Epoch 51/100
32/32 [==============================] - 4s 111ms/step - loss: 0.0633 - output_1_loss:
0.0318 - output_2_loss: 0.0316 - output_1_accuracy: 0.6290 - output_2_accuracy: 0.6315
Epoch 52/100
32/32 [==============================] - 4s 113ms/step - loss: 0.0634 - output_1_loss:
0.0319 - output_2_loss: 0.0315 - output_1_accuracy: 0.6300 - output_2_accuracy: 0.6306
Epoch 53/100
32/32 [==============================] - 4s 116ms/step - loss: 0.0634 - output_1_loss:
0.0316 - output_2_loss: 0.0317 - output_1_accuracy: 0.6355 - output_2_accuracy: 0.6342
Epoch 54/100
32/32 [==============================] - 4s 111ms/step - loss: 0.0630 - output_1_loss:
0.0315 - output_2_loss: 0.0315 - output_1_accuracy: 0.6301 - output_2_accuracy: 0.6370
Epoch 55/100
32/32 [==============================] - 4s 112ms/step - loss: 0.0632 - output_1_loss:
0.0316 - output_2_loss: 0.0315 - output_1_accuracy: 0.6391 - output_2_accuracy: 0.6368
Epoch 56/100
32/32 [==============================] - 3s 105ms/step - loss: 0.0633 - output_1_loss:
0.0316 - output_2_loss: 0.0316 - output_1_accuracy: 0.6301 - output_2_accuracy: 0.6361
Epoch 57/100
32/32 [==============================] - 3s 105ms/step - loss: 0.0634 - output_1_loss:
0.0317 - output_2_loss: 0.0317 - output_1_accuracy: 0.6270 - output_2_accuracy: 0.6361
Epoch 58/100
32/32 [==============================] - 3s 109ms/step - loss: 0.0631 - output_1_loss:
0.0318 - output_2_loss: 0.0313 - output_1_accuracy: 0.6266 - output_2_accuracy: 0.6344
Epoch 59/100
32/32 [==============================] - 3s 104ms/step - loss: 0.0630 - output_1_loss:
```

```
0.0315 - output_2_loss: 0.0315 - output_1_accuracy: 0.6339 - output_2_accuracy: 0.6419
Epoch 60/100
32/32 [==============================] - 3s 108ms/step - loss: 0.0631 - output_1_loss:
0.0313 - output_2_loss: 0.0318 - output_1_accuracy: 0.6335 - output_2_accuracy: 0.6391
Epoch 61/100
32/32 [==============================] - 4s 113ms/step - loss: 0.0630 - output_1_loss:
0.0315 - output_2_loss: 0.0315 - output_1_accuracy: 0.6363 - output_2_accuracy: 0.6385
Epoch 62/100
32/32 [==============================] - 3s 108ms/step - loss: 0.0629 - output_1_loss:
0.0314 - output_2_loss: 0.0316 - output_1_accuracy: 0.6352 - output_2_accuracy: 0.6370
Epoch 63/100
32/32 [==============================] - 3s 105ms/step - loss: 0.0630 - output_1_loss:
0.0316 - output_2_loss: 0.0314 - output_1_accuracy: 0.6358 - output_2_accuracy: 0.6374
Epoch 64/100
32/32 [==============================] - 4s 110ms/step - loss: 0.0627 - output_1_loss:
0.0313 - output_2_loss: 0.0315 - output_1_accuracy: 0.6379 - output_2_accuracy: 0.6364
Epoch 65/100
32/32 [==============================] - 4s 119ms/step - loss: 0.0630 - output_1_loss:
0.0314 - output_2_loss: 0.0316 - output_1_accuracy: 0.6301 - output_2_accuracy: 0.6384
Epoch 66/100
32/32 [==============================] - 4s 123ms/step - loss: 0.0630 - output_1_loss:
0.0315 - output_2_loss: 0.0315 - output_1_accuracy: 0.6328 - output_2_accuracy: 0.6391
Epoch 67/100
32/32 [==============================] - 4s 122ms/step - loss: 0.0629 - output_1_loss:
0.0314 - output_2_loss: 0.0314 - output_1_accuracy: 0.6386 - output_2_accuracy: 0.6351
Epoch 68/100
32/32 [==============================] - 4s 116ms/step - loss: 0.0627 - output_1_loss:
0.0312 - output_2_loss: 0.0315 - output_1_accuracy: 0.6315 - output_2_accuracy: 0.6318
Epoch 69/100
32/32 [==============================] - 4s 125ms/step - loss: 0.0627 - output_1_loss:
0.0314 - output_2_loss: 0.0313 - output_1_accuracy: 0.6323 - output_2_accuracy: 0.6353
Epoch 70/100
32/32 [==============================] - 4s 110ms/step - loss: 0.0629 - output_1_loss:
0.0315 - output_2_loss: 0.0314 - output_1_accuracy: 0.6350 - output_2_accuracy: 0.6411
Epoch 71/100
32/32 [==============================] - 4s 115ms/step - loss: 0.0628 - output_1_loss:
0.0312 - output_2_loss: 0.0316 - output_1_accuracy: 0.6323 - output_2_accuracy: 0.6384
Epoch 72/100
32/32 [==============================] - 3s 108ms/step - loss: 0.0625 - output_1_loss:
0.0311 - output_2_loss: 0.0314 - output_1_accuracy: 0.6356 - output_2_accuracy: 0.6376
Epoch 73/100
32/32 [==============================] - 4s 110ms/step - loss: 0.0621 - output_1_loss:
0.0309 - output_2_loss: 0.0312 - output_1_accuracy: 0.6393 - output_2_accuracy: 0.6433
Epoch 74/100
32/32 [==============================] - 3s 106ms/step - loss: 0.0627 - output_1_loss:
0.0313 - output_2_loss: 0.0314 - output_1_accuracy: 0.6361 - output_2_accuracy: 0.6302
Epoch 75/100
32/32 [==============================] - 4s 117ms/step - loss: 0.0626 - output_1_loss:
0.0313 - output_2_loss: 0.0312 - output_1_accuracy: 0.6364 - output_2_accuracy: 0.6400
Epoch 76/100
32/32 [==============================] - 4s 113ms/step - loss: 0.0626 - output_1_loss:
0.0313 - output_2_loss: 0.0313 - output_1_accuracy: 0.6349 - output_2_accuracy: 0.6351
Epoch 77/100
32/32 [==============================] - 3s 109ms/step - loss: 0.0628 - output_1_loss:
0.0311 - output_2_loss: 0.0317 - output_1_accuracy: 0.6291 - output_2_accuracy: 0.6316
Epoch 78/100
32/32 [==============================] - 3s 109ms/step - loss: 0.0622 - output_1_loss:
0.0312 - output_2_loss: 0.0311 - output_1_accuracy: 0.6302 - output_2_accuracy: 0.6349
Epoch 79/100
32/32 [==============================] - 4s 110ms/step - loss: 0.0627 - output_1_loss:
```

```
0.0313 - output_2_loss: 0.0315 - output_1_accuracy: 0.6343 - output_2_accuracy: 0.6331
Epoch 80/100
32/32 [==============================] - 4s 109ms/step - loss: 0.0623 - output_1_loss:
0.0312 - output_2_loss: 0.0311 - output_1_accuracy: 0.6372 - output_2_accuracy: 0.6362
Epoch 81/100
32/32 [==============================] - 4s 121ms/step - loss: 0.0623 - output_1_loss:
0.0310 - output_2_loss: 0.0313 - output_1_accuracy: 0.6336 - output_2_accuracy: 0.6384
Epoch 82/100
32/32 [==============================] - 3s 107ms/step - loss: 0.0622 - output_1_loss:
0.0311 - output_2_loss: 0.0311 - output_1_accuracy: 0.6317 - output_2_accuracy: 0.6407
Epoch 83/100
32/32 [==============================] - 3s 108ms/step - loss: 0.0621 - output_1_loss:
0.0310 - output_2_loss: 0.0310 - output_1_accuracy: 0.6349 - output_2_accuracy: 0.6367
Epoch 84/100
32/32 [==============================] - 3s 106ms/step - loss: 0.0622 - output_1_loss:
0.0311 - output_2_loss: 0.0311 - output_1_accuracy: 0.6308 - output_2_accuracy: 0.6328
Epoch 85/100
32/32 [==============================] - 3s 108ms/step - loss: 0.0622 - output_1_loss:
0.0311 - output_2_loss: 0.0311 - output_1_accuracy: 0.6352 - output_2_accuracy: 0.6358
Epoch 86/100
32/32 [==============================] - 3s 107ms/step - loss: 0.0622 - output_1_loss:
0.0310 - output_2_loss: 0.0312 - output_1_accuracy: 0.6323 - output_2_accuracy: 0.6309
Epoch 87/100
32/32 [==============================] - 3s 103ms/step - loss: 0.0621 - output_1_loss:
0.0310 - output_2_loss: 0.0310 - output_1_accuracy: 0.6353 - output_2_accuracy: 0.6380
Epoch 88/100
32/32 [==============================] - 4s 114ms/step - loss: 0.0619 - output_1_loss:
0.0308 - output_2_loss: 0.0310 - output_1_accuracy: 0.6362 - output_2_accuracy: 0.6342
Epoch 89/100
32/32 [==============================] - 4s 115ms/step - loss: 0.0618 - output_1_loss:
0.0308 - output_2_loss: 0.0310 - output_1_accuracy: 0.6295 - output_2_accuracy: 0.6382
Epoch 90/100
32/32 [==============================] - 4s 112ms/step - loss: 0.0622 - output_1_loss:
0.0312 - output_2_loss: 0.0310 - output_1_accuracy: 0.6268 - output_2_accuracy: 0.6339
Epoch 91/100
32/32 [==============================] - 3s 106ms/step - loss: 0.0620 - output_1_loss:
0.0309 - output_2_loss: 0.0311 - output_1_accuracy: 0.6287 - output_2_accuracy: 0.6384
Epoch 92/100
32/32 [==============================] - 3s 108ms/step - loss: 0.0624 - output_1_loss:
0.0314 - output_2_loss: 0.0311 - output_1_accuracy: 0.6280 - output_2_accuracy: 0.6309
Epoch 93/100
32/32 [==============================] - 3s 107ms/step - loss: 0.0619 - output_1_loss:
0.0308 - output_2_loss: 0.0311 - output_1_accuracy: 0.6361 - output_2_accuracy: 0.6377
Epoch 94/100
32/32 [==============================] - 3s 107ms/step - loss: 0.0617 - output_1_loss:
0.0308 - output_2_loss: 0.0308 - output_1_accuracy: 0.6354 - output_2_accuracy: 0.6349
Epoch 95/100
32/32 [==============================] - 4s 119ms/step - loss: 0.0622 - output_1_loss:
0.0313 - output_2_loss: 0.0309 - output_1_accuracy: 0.6353 - output_2_accuracy: 0.6375
Epoch 96/100
32/32 [==============================] - 4s 114ms/step - loss: 0.0619 - output_1_loss:
0.0310 - output_2_loss: 0.0309 - output_1_accuracy: 0.6354 - output_2_accuracy: 0.6398
Epoch 97/100
32/32 [==============================] - 3s 107ms/step - loss: 0.0624 - output_1_loss:
0.0314 - output_2_loss: 0.0310 - output_1_accuracy: 0.6333 - output_2_accuracy: 0.6351
Epoch 98/100
32/32 [==============================] - 3s 107ms/step - loss: 0.0617 - output_1_loss:
0.0307 - output_2_loss: 0.0309 - output_1_accuracy: 0.6280 - output_2_accuracy: 0.6379
Epoch 99/100
32/32 [==============================] - 4s 111ms/step - loss: 0.0617 - output_1_loss:
```

```
0.0309 - output_2_loss: 0.0308 - output_1_accuracy: 0.6313 - output_2_accuracy: 0.6423
Epoch 100/100
32/32 [==============================] - 4s 112ms/step - loss: 0.0617 - output_1_loss:
0.0308 - output_2_loss: 0.0309 - output_1_accuracy: 0.6328 - output_2_accuracy: 0.6319
```

Out[58]:
```
<keras.callbacks.History at 0x218d4e91130>
```

loss: 0.0617 - output_1_loss: 0.0308 - output_2_loss: 0.0309 - output_1_accuracy: 0.6328 - output_2_accuracy: 0.6319

In [ ]:

In [82]:
```python
latent_dim = 64
class Autoencoder4_1(Model):
    def __init__(self, latent_dim):
        super(Autoencoder4_1, self).__init__()
        self.latent_dim = latent_dim
        #Our encoder(using cnn)
        self.encoder = tf.keras.Sequential([
              layers.Conv2D(32, kernel_size=3, strides=1, padding='same', activation='r
              BatchNormalization(),
              layers.Conv2D(32, kernel_size=3, strides=2, padding='same', activation='r
              layers.Conv2D(32, kernel_size=3, strides=1, padding='same', activation='r
              BatchNormalization(),


        ])
        self.decoderOne = tf.keras.Sequential([
            layers.UpSampling2D((2, 2)),
            layers.Conv2D(32, kernel_size=3, strides=1, padding='same', activation='rel
             BatchNormalization(),
            layers.Conv2D(3,  kernel_size=1, strides=1, padding='same', activation='sig

        ])


    def call(self, x):
        encoded = self.encoder(x)
        decoded1 = self.decoderOne(encoded)
        return decoded1

autoencoder4_1= Autoencoder4_1(latent_dim)
```

In [83]:
```python
autoencoder4_1.compile(optimizer='adam',metrics=["accuracy"], loss=losses.MeanSquaredEr
```

In [84]:
```python
autoencoder4_1.fit(meanTrainingset,imageset1,epochs=50)
```

```
Epoch 1/50
32/32 [==============================] - 3s 61ms/step - loss: 0.0483 - accuracy: 0.4182
Epoch 2/50
32/32 [==============================] - 2s 61ms/step - loss: 0.0358 - accuracy: 0.5007
Epoch 3/50
32/32 [==============================] - 2s 62ms/step - loss: 0.0343 - accuracy: 0.5464
Epoch 4/50
32/32 [==============================] - 2s 64ms/step - loss: 0.0336 - accuracy: 0.5678
```

```
Epoch 5/50
32/32 [==============================] - 2s 63ms/step - loss: 0.0336 - accuracy: 0.5725
Epoch 6/50
32/32 [==============================] - 2s 65ms/step - loss: 0.0334 - accuracy: 0.5817
Epoch 7/50
32/32 [==============================] - 2s 62ms/step - loss: 0.0332 - accuracy: 0.5898
Epoch 8/50
32/32 [==============================] - 2s 62ms/step - loss: 0.0330 - accuracy: 0.5860
Epoch 9/50
32/32 [==============================] - 2s 61ms/step - loss: 0.0327 - accuracy: 0.5980
Epoch 10/50
32/32 [==============================] - 2s 61ms/step - loss: 0.0327 - accuracy: 0.5972
Epoch 11/50
32/32 [==============================] - 2s 62ms/step - loss: 0.0325 - accuracy: 0.6014
Epoch 12/50
32/32 [==============================] - 2s 62ms/step - loss: 0.0327 - accuracy: 0.6040
Epoch 13/50
32/32 [==============================] - 2s 62ms/step - loss: 0.0328 - accuracy: 0.6028
Epoch 14/50
32/32 [==============================] - 2s 62ms/step - loss: 0.0322 - accuracy: 0.6112
Epoch 15/50
32/32 [==============================] - 2s 62ms/step - loss: 0.0324 - accuracy: 0.6107
Epoch 16/50
32/32 [==============================] - 2s 63ms/step - loss: 0.0324 - accuracy: 0.6092
Epoch 17/50
32/32 [==============================] - 2s 67ms/step - loss: 0.0319 - accuracy: 0.6114
Epoch 18/50
32/32 [==============================] - 3s 80ms/step - loss: 0.0323 - accuracy: 0.6100
Epoch 19/50
32/32 [==============================] - 2s 77ms/step - loss: 0.0319 - accuracy: 0.6186
Epoch 20/50
32/32 [==============================] - 2s 77ms/step - loss: 0.0323 - accuracy: 0.6170
Epoch 21/50
32/32 [==============================] - 2s 76ms/step - loss: 0.0320 - accuracy: 0.6120
Epoch 22/50
32/32 [==============================] - 2s 75ms/step - loss: 0.0320 - accuracy: 0.6151
Epoch 23/50
32/32 [==============================] - 2s 77ms/step - loss: 0.0320 - accuracy: 0.6146
Epoch 24/50
32/32 [==============================] - 2s 77ms/step - loss: 0.0319 - accuracy: 0.6157
Epoch 25/50
32/32 [==============================] - 2s 75ms/step - loss: 0.0321 - accuracy: 0.6184
Epoch 26/50
32/32 [==============================] - 2s 75ms/step - loss: 0.0319 - accuracy: 0.6198
Epoch 27/50
32/32 [==============================] - 3s 82ms/step - loss: 0.0321 - accuracy: 0.6237
Epoch 28/50
32/32 [==============================] - 2s 77ms/step - loss: 0.0320 - accuracy: 0.6183
Epoch 29/50
32/32 [==============================] - 2s 76ms/step - loss: 0.0318 - accuracy: 0.6202
Epoch 30/50
32/32 [==============================] - 2s 76ms/step - loss: 0.0319 - accuracy: 0.6219
Epoch 31/50
32/32 [==============================] - 2s 77ms/step - loss: 0.0318 - accuracy: 0.6269
Epoch 32/50
32/32 [==============================] - 3s 79ms/step - loss: 0.0319 - accuracy: 0.6201
Epoch 33/50
32/32 [==============================] - 3s 81ms/step - loss: 0.0317 - accuracy: 0.6254
Epoch 34/50
32/32 [==============================] - 3s 94ms/step - loss: 0.0320 - accuracy: 0.6230
```

```
Epoch 35/50
32/32 [==============================] - 3s 94ms/step - loss: 0.0319 - accuracy: 0.6259
Epoch 36/50
32/32 [==============================] - 3s 79ms/step - loss: 0.0319 - accuracy: 0.6252
Epoch 37/50
32/32 [==============================] - 2s 76ms/step - loss: 0.0318 - accuracy: 0.6304
Epoch 38/50
32/32 [==============================] - 2s 75ms/step - loss: 0.0318 - accuracy: 0.6242
Epoch 39/50
32/32 [==============================] - 3s 81ms/step - loss: 0.0317 - accuracy: 0.6270
Epoch 40/50
32/32 [==============================] - 3s 88ms/step - loss: 0.0319 - accuracy: 0.6242
Epoch 41/50
32/32 [==============================] - 3s 86ms/step - loss: 0.0318 - accuracy: 0.6268
Epoch 42/50
32/32 [==============================] - 2s 78ms/step - loss: 0.0315 - accuracy: 0.6219
Epoch 43/50
32/32 [==============================] - 2s 77ms/step - loss: 0.0317 - accuracy: 0.6283
Epoch 44/50
32/32 [==============================] - 2s 75ms/step - loss: 0.0317 - accuracy: 0.6286
Epoch 45/50
32/32 [==============================] - 2s 78ms/step - loss: 0.0316 - accuracy: 0.6287
Epoch 46/50
32/32 [==============================] - 3s 81ms/step - loss: 0.0316 - accuracy: 0.6279
Epoch 47/50
32/32 [==============================] - 2s 75ms/step - loss: 0.0316 - accuracy: 0.6332
Epoch 48/50
32/32 [==============================] - 2s 76ms/step - loss: 0.0314 - accuracy: 0.6320
Epoch 49/50
32/32 [==============================] - 2s 78ms/step - loss: 0.0318 - accuracy: 0.6288
Epoch 50/50
32/32 [==============================] - 2s 76ms/step - loss: 0.0316 - accuracy: 0.6329
```

Out[84]:
```
<keras.callbacks.History at 0x218d0a8c040>
```

Testing dataset

In [60]:
```python
imageset1Test =  np.array([])
imageset2Test =  np.array([])
meanTrainingsetLabelTest = np.array([])
meanTrainingsetTest = np.array([])
```

In [61]:
```python
for i in range(500):
    x1 = random.randint(0, 25000)
    x2 = random.randint(25000,50000)
    imageset1Test = np.append(imageset1Test,x_train[x1])
    imageset2Test = np.append(imageset2Test,x_train[x2])
    x3 = (x_train[x1] + x_train[x2])/2
    x3Label =labels[int(y_train[x1])]+"." +labels[int(y_train[x2])]
    meanTrainingsetLabelTest = np.append(meanTrainingsetLabelTest, x3Label)
    meanTrainingsetTest = np.append(meanTrainingsetTest, x3)
```

In [62]:
```python
meanTrainingsetTest = meanTrainingsetTest.reshape(500,32,32,3)
imageset1Test = imageset1Test.reshape(500,32,32,3)
```

```
imageset2Test = imageset2Test.reshape(500,32,32,3)
```

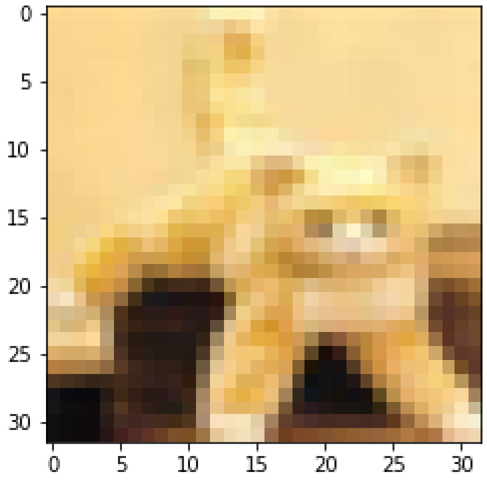
In [63]:
```
encoded_imgs  = autoencoder4.encoder(meanTrainingsetTest).numpy()
decoded_imgs1 = autoencoder4.decoderOne(encoded_imgs).numpy()
decoded_imgs2 = autoencoder4.decoderTwo(encoded_imgs).numpy()
```

In [64]:
```
decoded_imgs1.shape
```

Out[64]:
```
(500, 32, 32, 3)
```

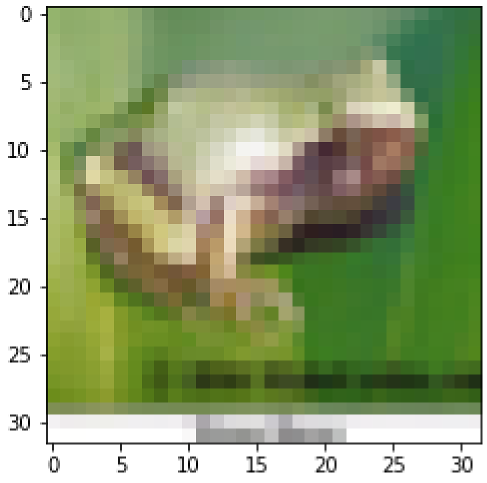
In [72]:
```
plt.imshow(imageset1Test[1])
print("cat")
```

```
cat
```



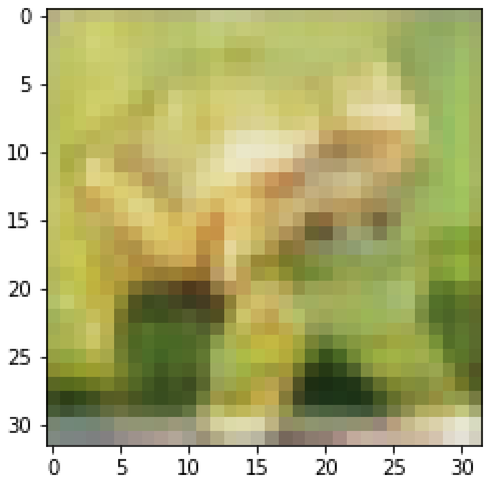
In [73]:
```
plt.imshow(imageset2Test[1])
print("frog")
```

```
frog
```



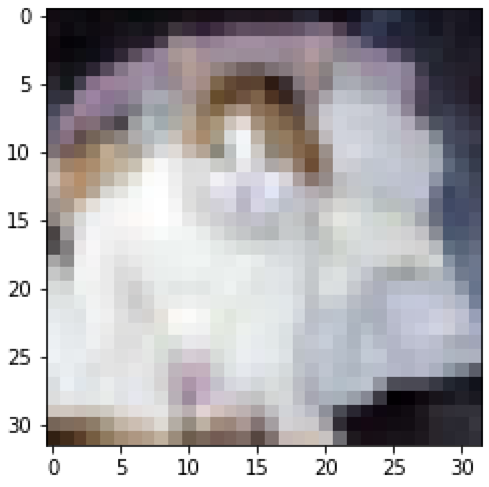
In [75]:
```
plt.imshow(decoded_imgs1[1])
```

Out[75]:
```
<matplotlib.image.AxesImage at 0x218d07c87f0>
```



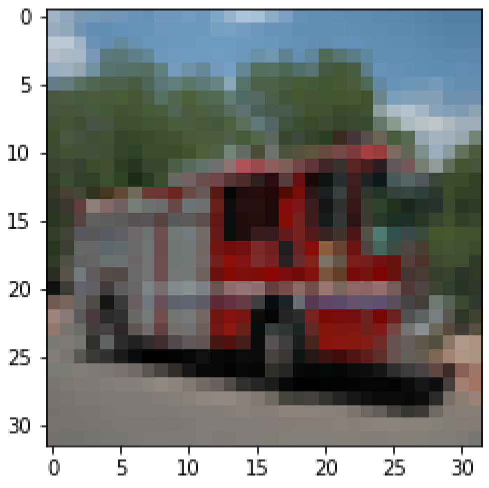
In [78]:
```
plt.imshow(imageset1Test[50])
```

Out[78]:
```
<matplotlib.image.AxesImage at 0x218d090c1c0>
```



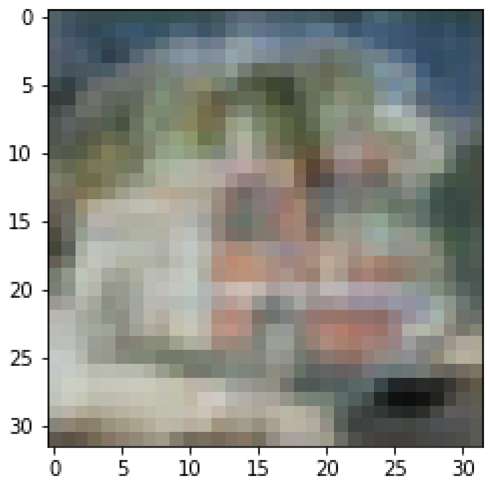
In [79]:
```
plt.imshow(imageset2Test[50])
```

Out[79]:
```
<matplotlib.image.AxesImage at 0x218d0967ee0>
```



In [80]:
```
plt.imshow(decoded_imgs1[50])
```

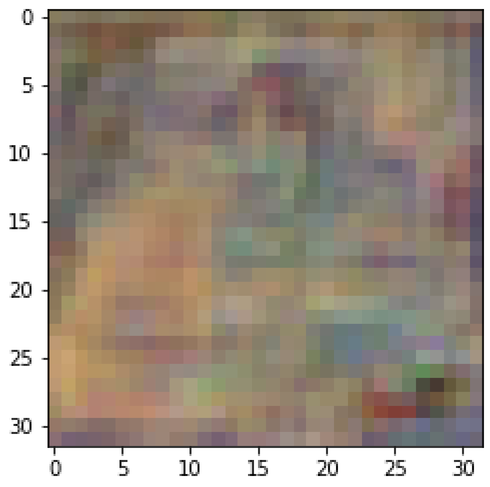
Out[80]:
```
<matplotlib.image.AxesImage at 0x218d09de100>
```



In [85]:
```
encoded_imgs_41  = autoencoder4_1.encoder(meanTrainingsetTest).numpy()
decoded_imgs1_41 = autoencoder4_1.decoderOne(encoded_imgs).numpy()
```

In [86]:
```
plt.imshow(decoded_imgs1_41[50])
```

Out[86]:
```
<matplotlib.image.AxesImage at 0x218d0c74130>
```



In [ ]:
```

```